Multimodale Interaktion in Virtueller Realität am Beispiel der virtuellen Konstruktion

Dissertationsschrift zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften

der Technischen Fakultät der Universität Bielefeld

vorgelegt von Marc Erich Latoschik am 31. Oktober 2000

<Titel>

<Bitte hier den Titel einfügen>

<Impressum>

Bitte nehmen Sie folgende Informationen in das Impressum auf:

Marc Erich Latoschik
Universität Bielefeld
Technische Fakultät
AG Wissensbasierte Systeme
Postfach 10 01 31
D-33501 Bielefeld

Von der Technischen Fakultät
der Universität Bielefeld
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
genehmigte Dissertation

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

Tag der Promotion: 28. Februar 2001

Kurztitel (bei Bedarf): *Multimodale Interaktion in Virtueller Realität*
oder kürzer: *Multimodale Interaktion in VR*

Geleitwort

Mit der Abkehr von üblicher Bildschirmausgabe stellt Virtuelle Realität (VR) den vermutlich stärksten Umbruch in der Multimediatechnologie dar. Unterstützt durch intelligente Funktionen soll VR beispielsweise im Fahrzeugversuchsbau ein "virtuelles Konstruieren" ermöglichen; man möchte Modelle realer Objekte bereits am CAD-basierten virtuellen Prototypen realistisch darstellen und explorieren können. Maus- und Tastatureingabe haben hier keinen Platz mehr. Verbesserte Formen der Mensch-Maschine-Interaktion sind erforderlich, die es gestatten, mit virtuellen Objekten ähnlich zu interagieren wie in der realen Umgebung.

In dem Schnittfeld von immersiver virtueller Realität und multimodaler Mensch-Maschine-Interaktion entwickelt Marc Latoschik in der vorliegenden Dissertation einen Ansatz, ein VR-System mit gesprochener Sprache, gekoppelt mit verschiedenen Formen der Gestikeingabe, zu steuern. Als technisches Resultat hat er im Rahmen des Projekts SGIM ("Sprach- und Gesteninterfaces für Multimedia") einen vollimplementierten Demonstrator entwickelt, der neben sprachbegleitenden Zeigegesten den Einsatz von sog. mimetischen ("vormachenden") Gesten zur kontinuierlichen Interaktion mit 3D-Objekten ermöglicht. Die Basis dafür sind sog. Aktuatoren, die die auflaufenden Kanal-Rohdaten synchronisieren und in Framezeit-basierten Teilstücken über im Szenengraph verankerte Knoten zur Auswertung an Bewegungsdetektoren übergeben. Die erhaltenen Bewegungsprimitive steuern über sog. Motion-Modifikatoren Objekt-Manipulatoren, mit denen die virtuellen Objekte gehandhabt werden. Die Erkennung signifikanter Muster in Merkmalssequenzen mit dem dafür entwickelten PrOSA-Verfahren ("Patterns on Sequences of Attributes") wird für Beispiele wie Greif- und Rotationsgesten verdeutlicht. Die schwierigste technische Aufgabe ist die bei der Zusammenführung von Sprache und Gestik herzustellende Korrespondenz zeitlich paralleler sprachlicher und gestischer Äußerungssegmente; sie werden über eine im Hinblick auf Zeitstempelverarbeitung erweiterte Form der Augmented Transition Networks (ATN) einander zugeordnet.

Die Erweiterung der gestischen und multimodalen Interaktionsformen ist speziell im Hinblick auf die mimetischen Gesten und ihre multimodale Integration ein wesentlicher Fortschritt, der unter Erhalt von (Quasi-)Standards den Anforderungen einer Echtzeit-VR-Umgebung standhält. Die Schriftfassung der Arbeit verdeutlicht zudem, daß der Autor eine immense Fülle an Konzeptquellen berücksichtigt hat und sich im Bereich der gegenwärtigen Virtual Reality wie auch der multimodalen Interaktion hervorragend ausweisen kann.

Bielefeld, im Mai 2001 Ipke Wachsmuth und Helge Ritter

Danksagung

Es stimmt ein wenig nachdenklich, dass nach einer Zeitspanne deren Beginn — so scheint es einem subjektiv — schon in weiter Ferne liegt, die Erinnerung an viele wichtige Personen und Ereignisse auf dem zurückliegenden Weg teilweise verblasst. Anscheinend nehmen uns die aktuellen Geschehnisse doch mehr in Anspruch, oder mit einem Achselzucken gesagt: "that's the wheel of time"?! Um sich dieser Ergebenheit nicht unterzuordnen, stelle ich hier all die Menschen in den Vordergrund, welche in der Vergangenheit meinen Weg gekreuzt und durch ihr aktives oder passives Dazutun mit die Einflüsse erzeugt haben, die letztendlich für meine Person auch in der Fertigstellung der vorliegenden Arbeit mündeten.

Die Mitarbeiter der Technischen Fakultät, insbesondere der AG WBS und der Gruppe M4I, haben durch ihre Kooperation und Freundschaft mehr als eine angenehme Arbeits- und Teamumgebung gebildet. Die immer fruchtbaren Diskussionen — ob über philosophische Unterschiede zwischen *realer* Welt und *diskreter* Modellbildung und Formen möglicher Berechnungsverfahren oder über den Einfluss der Trinkwasserfaktoren und der Brühdauer auf die Kaffee- und damit auf die Arbeitsqualität — waren echte Meilensteine im Tagesablauf.

Es fällt mir schwer, hier einzelne Personen zu nennen (vielleicht sollte ich netgroups angeben) oder eine Reihenfolge zu finden, so seht dieses nicht als Maß sondern als „Geste“: Dir Ipke, Stefan, Britta, Berni, Henrik, Sebastian, Frank, Timo, Christoph, Stefan, Claudia, Peter, Helge, Ian, Martin, Peter, Ralf, ... möchte ich danken. Meinen Freunden, die geglaubt haben ich sei verschollen, sende ich ein deutliches Ping! Danke an Kati, Peter & Co. für die Auszeiten (und diverse Manuskriptkorrekturen). Allen Liesen, keiner weiß das besser als Ihr: sunset now!

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Motivation 2
1.1.1 Mensch-Maschine-Kommunikation 2
1.1.2 Simulation und Immersion in der VR 5
1.1.3 Neue Interaktionsmethoden 7
1.2 Zielsetzung 10
1.3 Überblick 13

2 Stand der Forschung **15**
2.1 Multimodale Eingabesysteme in der Computergraphik 16
2.2 Gestik und Sprache 22
2.2.1 Phänomenologie gestischer Äußerung 22
2.2.2 Gestenklassifikationsschemata 26
2.2.3 Korrespondenz multimodaler Äußerungen 30
2.2.4 Verfahren maschineller Spracherkennung 31
2.2.5 Verfahren maschineller Gestenerkennung 34
2.3 Virtuelle Realität 36
2.3.1 Virtuelle Konstruktion 44
2.3.2 Interaktionstypen und Manipulationsanforderungen . . 46
2.4 Diskussion 48
2.4.1 Möglichkeiten sprach-gestischer Interaktionen 49
2.4.2 Gesten für eine multimodale Interaktion 50
2.4.3 Methoden zur Gestenerkennung 54
2.4.4 Multimodale Eingaben in VR 56

3 Basis der Interaktionsgestaltung **59**
3.1 Mögliche Interaktions-Gestenformen 59
3.2 Diskrete und kontinuierliche Interaktionsabläufe 61
3.3 Aufgabenstellung 64

4 Ein allgemeiner Ansatz zur Gestenerkennung in der VR 67
4.1 Gestenmerkmale und ihre Verankerung 67
4.1.1 Bezugssysteme der Interaktionsgestik 67
4.1.2 Spatiotemporale Merkmale der Interaktionsgestik . . . 70
4.1.3 Sensorik 74
4.1.4 Anforderungen an das Gestenerkennungs- und Analysesystem 76
4.2 Aktuatoren 80
4.2.1 Konzeptbasis 80
4.2.2 Definition des Abstrakten Aktuators 83
4.2.3 Aktuatorknoten 90
4.2.4 Aktuator- und Sequenzfeldtypen 95
4.2.5 Aktuatoren vs. Knoteninstanzen 108
4.3 Gestenerkennung mit PrOSA 110
4.3.1 Detektoren 113
4.3.2 Event- vs. Intervallergebnis 117
4.3.3 Verschaltungen von Detektoren 120
4.4 Motion-Modifikatoren 123
4.4.1 Constraintmatrix 124
4.4.2 Definition der Motion-Modifikatoren 126
4.4.3 Ein Rotations-Motion-Modifikator 128
4.4.4 Manipulatoren 131
4.5 Zusammenfassung 132

5 ATN-basierte Auswertung multimodaler Interaktionen 135
5.1 Analyse sprachlicher Handlungsanweisungen 135
5.1.1 Exemplarische verbale Äußerungen 135
5.1.2 Struktur und Semantik der Handlungsanweisungen . . 137
5.2 Sprachgestützte Gestik - Multimodalität 140
5.2.1 Semantische Zusammenhänge 141
5.3 Ein erweitertes ATN für die multimodale Integration 142
5.3.1 Voraussetzungen 142
5.3.2 Methodenübersicht 143
5.3.3 ATN-Modellierung 144
5.4 Äußerungsinterpretation und -umsetzung 151
5.4.1 Ein Ansatz zur multimodalen Referenzanalyse 151
5.4.2 Auswertung gestischer Referenzinformation 155
5.4.3 Manipulationsumsetzung 163
5.5 Zusammenfassung 165

6 SGIM - eine exemplarische Anwendung in der Virtuellen Konstruktion 167
6.1 Aufbau und Funktionsbeschreibung 167
6.2 Wahl des Modellierungswerkzeugs 169
6.3 Umsetzung der Konzepte im Prototyp 171
6.3.1 Ein externer Handformdetektor 171
6.3.2 Interne PrOSA-Detektoren 174
6.3.3 Implementation der VR-Konzepte 176
6.4 Beispielinteraktionen 178
6.4.1 Das virtuelle Labor 178
6.4.2 Objekterzeugung 180
6.4.3 Selektion von Objekten 180
6.4.4 Verschieben 181
6.4.5 Rotieren 181
6.4.6 Verbindungen diskret etablieren 182
6.4.7 Verbindungen kontinuierlich etablieren 183
6.5 Zusammenfassung 183

7 Resümee 185
7.1 Ergebnisse 186
7.2 Diskussion 189
7.3 Ausblick 190

Abbildungsverzeichnis

1.1 WIMP-basierte Mensch-Maschine-Interaktion 3
1.2 Immersion vor einer *Wall* 6
1.3 Interaktion an einer *Workbench* 8
1.4 Betrachtete Teilgebiete 10

2.1 Gestenphasen 24

4.1 Referenzstrahlenverankerung 69
4.2 Kanalsynchronisation 87
4.3 Synchronisation Sigma 1 88
4.4 Synchronisation Sigma 2 89
4.5 Synchronisation Sigma 3 89
4.6 Aktuatorknoten 91
4.7 Handformklassifikationsfunktion 97
4.8 Aktuatortransformation 102
4.9 Deixisklassifikationsfunktion 105
4.10 Deixisklassifikationsfunktion mit Cosinusverlauf 106
4.11 Aktuatoren vs. Aktuatorknoten 108
4.12 Singulärer Detektor 116
4.13 Schwellwertüberprüfung 118
4.14 Gestenerkennung durch hierarchische Detektorenanordnung . 122
4.15 Plot einer Kreisbewegung 124
4.16 Aktivierungsalgorithmus eines Motion-Modifikators 128
4.17 Bewegungsabstraktion durch einen Motion-Modifikator 130
4.18 Teilmanipulationen durch einen Motion-Modifikator 131
4.19 Manipulator während einer Teilrotation 132

5.1 Multimodale Anwendungslogik als ATN 146
5.2 Top-Level-ATN für die Interaktionserkennung 148
5.3 ATN Traversierungsalgorithmus 149

5.4 Teil-ATN für Objektbeschreibungen durch DNPs 152
5.5 Auswertealgorithmus der Referenzanalyse 154
5.6 Einbettung der Konzepte in eine VR-Anwendung 156
5.7 Referenzstrahlenfokus und Abstandsermittlung 157
5.8 Spacemap eines Referenzstrahls 160
5.9 Spacemaps für drei Referenzstrahlen 162
5.10 Zusammenwirken von Aktuator, Motion-Modifikator und Manipulator 164

6.1 Graphisches Erkennerfrontend 173
6.2 Externe eventgetriebene Gestendetektionsmodule 174
6.3 Einbettung eng gekoppelter Gestendetektionsmodule 175
6.4 Eintritt in das virtuelle Labor 178
6.5 Blick in das virtuelle Labor 179
6.6 Navigation in der Szene 179
6.7 Verbale Objektinstanziierung 180
6.8 Multimodale Selektion 180
6.9 Verschiebung eines Objekts 181
6.10 Kontinuierliche Rotationsvermittlung 181
6.11 Diskreter Verbindungsschluss 182
6.12 Interaktives Verbinden durch kontinuierliches Heranführen . . 183

Symbolverzeichnis

$[\ldots]$	Optionaler Eintrag
?, wie in “was?”	Notationsstil für Abfragefunktionen
*, wie in “*END”	ATN look-ahead
${}_2A$	Abstrakter Aktuator
$\mathcal{A}(t)$,$\mathcal{A}_l(t)$,$A(t)$	Attributsequenzen
α	Analysefunktion
AK	Aktuatorknoten
β	Berechnungsfunktion
$\delta_p(i, [A_k])$	Kontinuierliche Mappingfunktion
C,$\mathcal{C}$	Constraintmatrix (oder Menge von Kinderknoten)
$\vec{\delta}_j(t)$	Diskrete Wertzugriffsfunktion
e,e_j	Eingabekanäle
$FM(t)$	Menge von Ausschnitten eines Aktuators
GD	Detektor
$\mathcal{H}$	Menge von Testprädikaten
i_x	Geschlossenes Zeitintervall
$\mathcal{K}_\mathcal{E}$	Menge von Eingabekanälen
$\mathcal{E}_j$	Kanaleingaberaum
$\vec{\kappa}_j(t)$	Kontinuierliche Wertzugriffsfunktion
M	Homogene Transformationsmatrix
$\mu(A_i, i)$	Initiale Mappingfunktion
$\mathcal{N}$	Normierungsfunktion
ω	Gewichtungsfaktor oder Bewertungsfunktion
p_x	Property
ϕ	Subsamplefunktion
r	Wiederholrate, Frequenz
$\mathcal{R}$	Menge von vorgegebenen (Raster-)Vektoren
$\rho(t')$	Logische Verknüpfung von Testbedingungen

$S_j(t)$	Menge von Samplewerten
σ,σ_x	Synchronisationsfunktionen
t,t_j,t^{min},t^{max},t_x^j	Zeitpunkte
$T_j(t)$	Menge von Zeitpunkten
$T_{eval}(t)$	Traversierungszeitpunkte in der Render-Loop
T_{c1},T_x	Transformationen
${}_xT^y$	Transformation von x nach y

Kapitel 1

Einleitung

Multimediale Virtuelle Umgebungen (VU) stellen ein neues Medium in der Mensch-Maschine-Kommunikation (MMK) dar. Durch die Präsenz und Einbettung des Benutzers im Arbeitsbereich der Anwendung erscheinen bisherige Interaktionsansätze und Eingabegeräte unnatürlich oder auf Grund ihrer engen Verknüpfung mit bisherigen Computerarbeitsplätzen nicht mehr adäquat. Hier sollen Alternativen zu herkömmlichen Interaktionsformen im Rahmen einer Anwendung zur Virtuellen Konstruktion erprobt werden. Die vorliegende Arbeit dokumentiert anhand der Entwicklung von Kernkomponenten des SGIM[1]-Systems (Speech and Gesture Interfaces for Multimedia), einer Anwendung für die sprach-gestische Steuerung in der Virtuellen Konstruktion (VK), die Ergebnisse bei der Erforschung neuer Eingabemetaphern in einer speziellen Anwendung der Virtuellen Realität (VR). Untersucht werden in diesem Kontext zum einen verschiedene Möglichkeiten eines Benutzers seine Interaktionswünsche multimodal auszudrücken, zum anderen die daraus resultierenden Anforderungen an die Verarbeitung der sprach-gestischen Anwender-Eingabedaten und deren konzeptuelle Einbettung in die speziellen Anforderungen einer VR-Anwendung.

Teile der in dieser Arbeit entwickelten Konzepte und Ergebnisse wurden vorpubliziert (s. [40], [41], [49], [50], [51], [52], [53], [86] und [96] im Literaturverzeichnis).

[1] Die vorliegende Arbeit entstand im Kontext des durch das Land Nordrhein-Westfalen geförderten Teilprojekts SGIM der Virtuellen Wissensfabrik.

1.1 Motivation

Um die Entwicklung neuer Interaktionsmethoden in einer speziellen Domäne zu motivieren, erfolgt einleitend eine allgemeine Betrachtung bisheriger Strömungen in der Etablierung einer Mensch-Maschine-Kommunikation sowie eine kritische Betrachtung der damit bisher eingesetzten Operations- und Eingabeformen. Darauf aufbauend werden neue Möglichkeiten der Interaktion mit Hilfe natürlicher menschlicher Kommunikationsmittel als Alternativen vorgeschlagen sowie die in diesem Kontext benötigten Begriffe eingeführt.

Die Beantwortung der Frage, warum diese neuen Methoden zur Manipulation und Interaktion speziell in Virtuellen Umgebungen benötigt werden, kann nicht ohne ein Verständnis über die grundlegenden Unterschiede dieser Umgebungen zu herkömmlichen Computerarbeitsplätzen und deren Interaktionsmöglichkeiten erfolgen. Daher werden anschließend die wesentlichen Eigenschaften Virtueller Umgebungen beschrieben und weiterhin sowohl die bisher eingesetzten, als auch neuere Ansätze für Interaktionstechniken in der Virtuellen Realität vorgestellt. Dieses spannt zum einen den Rahmen für die folgenden konzeptuellen Ausarbeitungen neuer Interaktionsmethoden auf, zum anderen wird so ein Einblick in die Adäquatheit dieser Methoden im Kontext Virtueller Umgebungen gegeben.

1.1.1 Mensch-Maschine-Kommunikation

Die vorliegende Arbeit steht im Kontext von Arbeiten zur Erforschung von Methoden der Mensch-Maschine-Kommunikation. Grundlegendes Ziel ist es hier, die bisherige Art und Weise der Maschinen*bedienung* durch *Interaktionsformen* zu ersetzen, die auf den Menschen und seine Kommunikationsmöglichkeiten zugeschnitten sind.

Handelt es sich um eine reine Bedienung, so wird von einem Anwender vorausgesetzt, die von der Maschine zur Verfügung gestellten restriktiven Eingabemöglichkeiten anzuwenden, um gewünschte Operationen durchzuführen. Dieses bedingt bei einem Großteil der bisher entwickelten technischen Systeme — vom einfachen Videorekorder bis zu elaborierteren Computersystemen — in hohem Maße die Akzeptanz des Benutzers, sich das formale Bedienungsprinzip der Maschine anzueignen. Ein solches Vorgehen verlangt von einem Anwender bereits implizit die Bereitstellung *kognitiver Kapazitäten* allein für die Bedienung des Systems. Um sich bewusst mit der Lösung einer Aufgabe zu befassen, muss ein Teil der geistigen und körperlichen Kompetenzen rein für die Erlernung der möglichen Operationsstruktu-

ren und Eingabeformen aufgewendet werden (s. Abbildung 1.1). Die dafür aufzubringende Kapazität steht der eigentlichen Lösungsfindung bei der zu bewältigenden Aufgabe nicht mehr zur Verfügung.

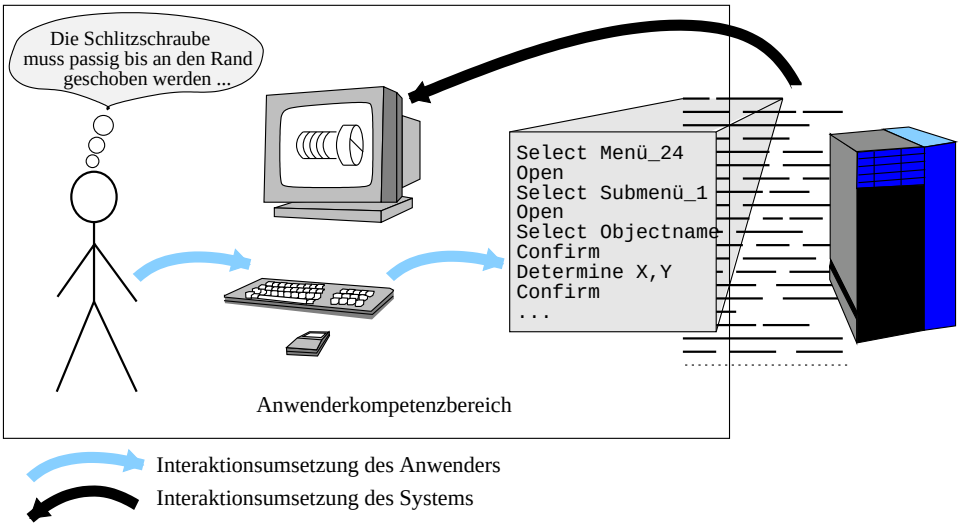

Abbildung 1.1: WIMP-basierte (s. Text) MMK. Ein Anwender benutzt die zur Verfügung stehenden Eingabegeräte, um seinen Interaktionswunsch in eine der vielen möglichen Eingabesequenzen im Formalismus des Systems umzusetzen.

Mit der Entwicklung immer komplexerer Computeranwendungen hat die Diskrepanz zwischen möglicher Funktionalität und dem damit verbundenen Lernaufwand, um sich dieser Funktionen zu bedienen, erheblich zugenommen. Die extrem gestiegenen Rechenleistungen der Maschinen haben in hohem Maße zur Bereitstellung immer umfangreicherer und vom funktionellen Standpunkt aus mächtigerer Programmpakete geführt, deren Bedienung findet dagegen nach wie vor mit Hilfe herkömmlicher Interaktionsmethoden (s.u.) statt. Diese haben sich parallel zu den jeweiligen technischen Möglichkeiten der Ein- und Ausgabegenerierung entwickelt. Angefangen mit veränderten Eingabearten, von manuellen Registereinstellungen über Lochkarten und keyboardbasierten Kommandosequenzen, fand ebenso eine Veränderung der Ausgabemöglichkeiten statt: begonnen mit Lichtsignalen, über Typenraddrucker hin zu Textbildschirmen, hat sich ebenfalls die Interaktionsform mit diesen Systemen gewandelt. Wurden sie ursprünglich in einem rein sequentiellen Batch- und später auch Kommandomodus betrieben, führte die Entwicklung 2D-graphikfähiger Ausgabegeräte (2D-Displays) — von Bild-

schirmen mit Pixel- oder Vektor-genauer Ansteuerung — zu neuen Interaktionsformen: Die heutzutage gängigen Systeme versuchen zu einem grossen Teil eine an einen realen Schreibtisch angelehnte Oberfläche anzubieten, indem einem Anwender ein computergeneriertes Abbild eines entsprechenden Arbeitsplatzes dargeboten wird. Kern dieser Art der Benutzerschnittstelle sind die Konzepte der *Interaktivität* und der *graphischen Repräsentation.* Sie ermöglichten die Entwicklung der sogenannten WIMP-Metaphern (WIMP: Windows, Icons, Menu, Pointers)[2]. Zentraler Bestandteil dieser ist eine interaktive Manipulation graphisch dargestellter Entitäten mit Hilfe von speziellen Eingabegeräten wie Maus, Trackball oder Joystick. Ausgabe- und Eingabefreiheitsgrade sind hier nahezu identisch. Die Eingabegeräte erlauben die Positionierung 2D-graphisch dargestellter Repräsentanten (den Pointern) auf den Ausgabegeräten. Zusätzliche Eingabeschalter gestatten es mit Hilfe der Eingabegeräte bestimmte Ereignisse auszulösen, welche sich kontextabhängig an den unter dem Pointer befindlichen Window, Icon oder Menu orientieren. Diese Form der Aktionsauswahl wird als *point-and-click* bezeichnet. So kann mit einem Mausclick ein Fenster geschlossen, eine mit einem Icon identifizierte Anwendung gestartet oder ein entsprechendes Untermenü aufgeklappt werden.

Die dargebotenen Interaktionsmöglichkeiten folgen dabei nur zu einem Teil bestimmten Standards — den sogenannten *Styleguides* — zur Bereitstellung von wiederkehrenden Basisoperationen wie etwa dem Laden und Speichern von veränderlichen Daten. Je nach Anwendungsgebiet können darüberhinaus völlig unterschiedliche Aktionen bereitgestellt und diese wiederum in etlichen Formen in den Menüs verankert werden.

Durch die zur Verfügung stehenden 2D-Displays wurde es insbesondere möglich, Anwendungen zur Manipulation graphischer Inhalte, etwa digitaler Bilder und technischer Zeichnungen u.Ä., zu entwickeln. Die Interaktion mit diesen Systemen verläuft größtenteils objektzentriert: Ausgewählte Objekte bestimmen die weiter zur Verfügung stehenden Operationen. Durch die Anzahl verschiedener Objekt- und Manipulationstypen wird dabei eine sehr breite Palette vieler verschiedener Operationen benötigt. Um diese einem Anwender zugänglich zu machen, werden bereits jetzt teilweise völlig überladene Menüstrukturen eingesetzt, um alle in den Systemen möglichen Funktionen in den WIMP-Interfaces zu verankern. Gilt dieses schon für 2D-graphische Anwendungen, so in besonderem Maße für die Manipulation von 3D-Objekten, z.Bsp. in Anwendungen aus dem CAD (Computer Aided Design)-Bereich oder in der Domäne des *Virtual Design.* Die dritte Dimen-

[2]Grob übersetzt als: Fenster, Ikonen, Menü, Zeiger.

sion wird hier auf 2D-Displays nur unzureichend emuliert (mittels 3D→2D Projektion, s. Abschnitt 2.3). Neben der Vielzahl an möglichen Operationen gilt es dabei zusätzlich einen weiteren manipulierbaren Freiheitsgrad zu bewältigen, dessen Parameter nicht explizit durch ein entsprechendes Eingabegerät zur Modifikation bereitgestellt wird.

Als Alternative zur Erlernung einer — eventuell unüberschaubaren — Menge vom System vorgegebener Operationen wird hier die Bereitstellung einer auf den menschlichen Kommunikationsmöglichkeiten basierenden Interaktion angestrebt. Die Entwicklung einer multimodalen Eingabe, konkret die Auswertung natürlicher sprachlicher und gestischer Äußerungen eines Benutzers, versucht, den kognitiven Lernungsaufwand des Anwenders auf die maschinelle Berechnungsseite zu verlagern. Anstatt immer mehr Rechenkapazität für neue Operationen zu reservieren, werden sie für die Bereitstellung *interpretativer Komponenten* verwendet. Basis ist hier nicht mehr — wie bisher häufig — eine vom System vorgegebene restriktive Eingabeform, sondern eine von starren Ablaufstrukturen befreite Kommunikation zwischen Mensch und Maschine. Sprache und Gestik erlauben es auf eine intuitive Art Objekte und Operationen zu spezifizieren sowie Manipulationen durch gestische Körperbewegungen in vier Dimensionen zu steuern. Hier wird die Annahme gemacht, dass eine solche Form der MMK die Interaktion mit den Systemen intuitiver, robuster und effizienter gestaltet.

1.1.2 Simulation und Immersion in der VR

Anwendungen der Virtuellen Realität basieren in besonderem Maße auf der Interaktion mit dreidimensionalen Objekten. Als primäre Eigenschaft vieler VR-gestützter Anwendungen wird dabei der *Immersionsfaktor* angesehen: Durch die Generierung, Simulation und Präsentation natürlich wirkender Umgebungen und Szenen wird einem Benutzer das Gefühl vermittelt, diese Szenen seien echt und Teil seiner bewussten und erfahrbaren Realität. Abbildung 1.2 (S. 6) veranschaulicht dieses am Beispiel einer immersiven Großprojektion einer Virtuellen Umgebung. Der Benutzer scheint hier umgeben von den zu manipulierenden virtuellen Objekten.

Um dieses zu erreichen, wurden bisher unterschiedliche Forschungsbestrebungen verfolgt. Legt man die Strukturierung klassischer computergestützter Systeme mit ihrer klaren Unterteilung in Input- und Outputbereiche zugrunde, so wurden in der VR-Vergangenheit etliche Bestrebungen unternommen, die Ausgabequalität zu perfektionieren. In erster Linie trieb man die Verbesserung der visuellen Qualität der Simulationen voran, da von den menschlichen Sinnen der optische als der wichtigste akzeptiert wird, um ei-

Abbildung 1.2: Immersion durch Head-Tracking und Stereoskopie an einer *Wall*, einem einseitigen Projektionsdisplay für VR-Anwendungen.

ne ausgeprägte Immersion zu erreichen. Primär werden die Szenen in der VR dreidimensional präsentiert und vermitteln einem Benutzer so das Gefühl der Tiefenpräsenz der Objekte; bisher geläufige computergraphische Stereoskopieverfahren erzeugen dafür für jedes Auge spezifische, der relativen Augenposition entsprechende Bilder, welche anschließend vom menschlichen Gehirn zu einer 3D-Repräsentation integriert werden.

Folgende Kriterien und Anforderungen sind dabei für eine realistisch erscheinende visuelle Simulation und Präsentation zu nennen: Die Perspektive, als mathematische Beschreibung der benutzerspezifischen Sicht gegeben durch die relative Lage des Sehapparates in einer Szene, sollte der Perspektive in der Realität entsprechen. Weiterhin wird eine Framerate — die Anzahl der präsentierten Bilder pro Sekunde — von mindestens 20Hz benötigt. Und letztendlich lässt sich fordern, dass die dargestellten Objekte in ihren simulierten Eigenschaften der Struktur, Farbe und Beleuchtung den in der Realität zu findenden Objekten in diesen Attributen gleichen. Um diese Vor-

aussetzungen zu erfüllen, wurden verschiedene Techniken entwickelt. Die Generierung von Stereoskopieeffekten durch Shutterbrillen oder Head-mounted Displays, Laserhologramme, diverse Headtrackingmethoden zur Ermittlung der Kopfposition und -lage, unterschiedliche Render- oder Shadingalgorithmen, der Einsatz von Texturen und diverse Optimierungen der Geometriespezifikation sollen exemplarisch hier genannt werden. Diese Ausführungen werden in Kapitel 2.3 konkretisiert; im Zuge dieser einleitenden Übersicht bleibt festzuhalten, dass die Voraussetzungen für eine visuelle Simulation, welche eine der Kernkomponenten eines VR-Systems darstellt, als gegeben anzusehen sind. Weitergehende Informationen sind der Literatur zu entnehmen (aktuell in den SIGGRAPH[3] Proceedings oder klassische Arbeiten in [10][19][25]).

In der Entwicklung der VR folgen Verbesserungen in der Generierung natürlicher oder naturgetreuer Eindrücke für weitere menschliche Sinne. So dient auch das Soundrendering, die realitätsnahe Generierung von auditorischen Eindrücken, der Verbesserung des Immersionsfaktors. Auch gibt es vermehrt Bestrebungen, unserem Tastsinn über haptisches Feedback den Eindruck zu vermitteln, mit realen Objekten in Kontakt zu stehen. Diese Erläuterungen veranschaulichen, welches breite Spektrum von Arbeiten und Bestrebungen besonders auf der Generierungsseite, dem Output eines Systems, sowohl in Qualität als auch in Quantität bereits erforscht wurde oder aktuell bearbeitet wird.

1.1.3 Neue Interaktionsmethoden

Input/Output - die Perfektionierung von Letzterem stellte bisher weitestgehend den Mittelpunkt der VR-Forschung dar und hat ihr Gesicht geprägt. Immersion — als eine der primären Eigenschaften der VR — bedingt aber ebenfalls eine möglichst natürliche Eingabe und Interaktion mit einem auf VR-Methoden basierenden System und den dort präsentierten Szenerien; dazu wird eine Methode benötigt, um dem Benutzer auch auf dieser Ebene das Gefühl einer realen Einbettung in die simulierte Umgebung zu vermitteln. Offensichtlich stellen bisherige bildschirmorientierte Eingabesysteme wie Maus und Tastatur keine adäquate Schnittstelle für solche Anwendungen dar, in denen der Benutzer frei stehend in oder vor einer simulierten Szene agiert. Sehr viele auf einer Großbildprojektion basierende Systeme, die vermehrte Benutzung einer "Wall" (Abbildung 1.2), einer "Workbench" [48] (Abbildung 1.3) oder einer "Cave" [19], zeichnet eine derartige platzungebundene Anwenderposition aus. Dagegen wurden viele der bisherigen VR-

[3] Special Interest Group on Computer Graphics der Association for Computing Machinery (ACM).

Eingabemethoden zu einem erheblichen Teil den Metaphern der 2D WIMP-Interfaces entlehnt (s. z.Bsp. Interaktionstechniken in [68]). Resultierend transportieren die Entwicklungen vieler der in der VR eingesetzten Eingabegeräte — die Pointingdevices wie 3D-Maus oder Stylus — das Prinzip von *point-and-click* in die Virtuellen Umgebungen: Die Position und Ausrichtung der Eingabegeräte wird nach Betätigung eines realen Knopfes ausgelesen. Dieser diskrete Eingabevektor dient daraufhin der Ermittlung eventuell getroffener, im 3D-Raum liegender Schalter und Menüs.

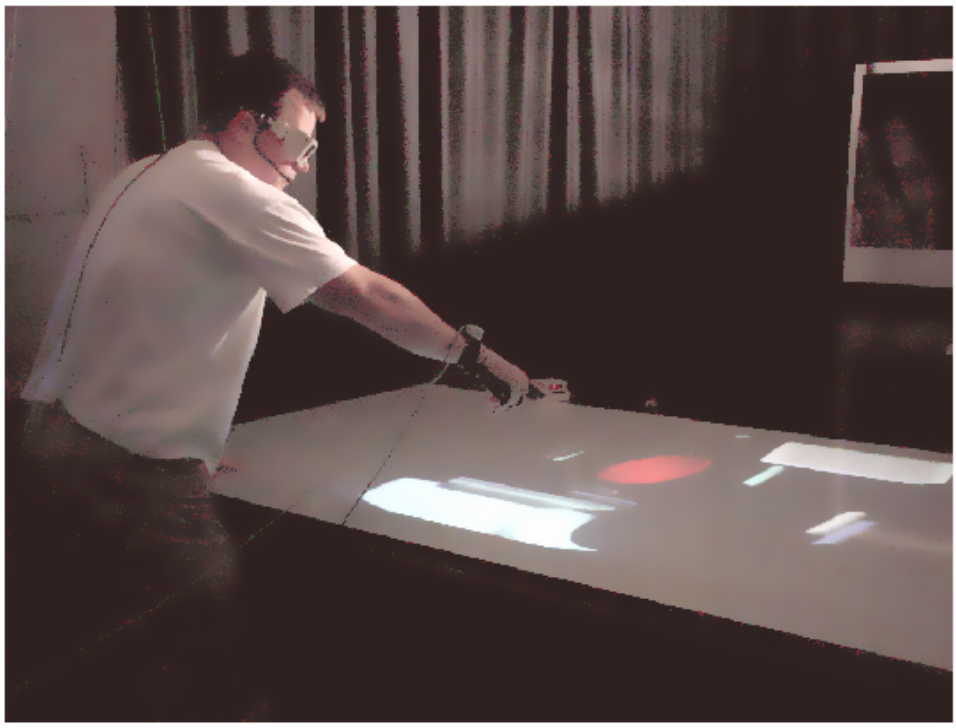

Abbildung 1.3: Neuartige Interaktion mit einer VR-Anwendung auf einer *Workbench*, einem an eine Werkbankmetapher angelehnten Großdisplay mit horizontaler Projektionsfläche. Statt spezieller Eingabedevices benutzt ein Anwender Sprache und Gestik, um gewünschte Veränderungen und Manipulationen auszulösen und durchzuführen (Das Bild zeigt bereits in einer Vorschau die in dieser Arbeit konzipierte multimodale Interfacekomponente bei einem Einsatz im SGIM-System).

Im Gegensatz dazu eröffnen die verschiedenen in der VR entwickelten Sensoriken grundlegende technische Möglichkeiten, die Körperbewegungen

eines Benutzers gänzlich oder teilweise zu erfassen. Diese auf dem elektromagnetischen, akustischen, gyroskopischen oder optischen Prinzip arbeitenden Verfahren liefern in einem quasi-kontinuierlichen Strom die Position und Ausrichtung verschiedener Fixpunkte und Referenzkoordinatensysteme auf dem Körper des Benutzers. Sie erlauben so prinzipiell auf Rohdatenbasis die Gestik, die *Körpersprache* eines Anwenders, als Eingabemodalität nutzbar zu machen. Bisher werden diese technischen Verfahren aber keineswegs in ihrem vollen Potential ausgeschöpft, sondern es erfolgt größtenteils nur eine Auswertung in begrenztem Maße, z.Bsp. für die Bereitstellung einfacher, an bisherige Mausinterfaces angelehnte Eingabeformen.

Gerade die Kombination mit den nun zur Verfügung stehenden Spracherkennungssystemen bietet dagegen ein hohes Potential, um eine natürliche — weitgehend von Eingabegeräten ungebundene — multimodale Interaktion zu realisieren, wie es in Abbildung 1.3 bereits angedeutet wird. Hier wird mit dem Einsatz neuer, auf sprach-gestischer Eingabe basierender MMK-Methoden, ein Vorschlag zur Bereitstellung einer intuitiveren, robusteren und effizienteren Interaktion gemacht.

Die Realisierung einer solchen multimodalen Schnittstelle für VR-Anwendungen bedingt die Auseinandersetzung mit unterschiedlichen Teildisziplinen. Es muss zuerst die Art der zu gestaltenden Interaktionen analysiert werden. Daraufhin stellt sich die Frage nach grundlegend einzusetzenden Verfahren zur Gesten- und Spracherkennung sowie zur multimodalen Integration und Interpretation. Und schliesslich müssen die erarbeiteten Konzepte unter den besonderen Rahmenbedingungen eines Einsatzes in der VR tragfähig sein. Zur Bearbeitung der damit verbundenen Problemstellungen werden in der vorliegenden Arbeit — und mit den darin entwickelten Ansätzen — weitgehend allgemeingültige Lösungsvorschläge unterbreitet und diese anhand einer exemplarischen Implementation validiert.

1.2 Zielsetzung

Um das heterogene Aufgabenspektrum zu bearbeiten, werden zunächst die einzelnen Teilaufgaben näher betrachtet. Das Ziel ist die Erforschung von Methoden zur multimodalen Interaktion in Virtuellen Umgebungen. Diese sollen konkret an einer Umsetzung erprobt werden: Der Steuerung einer Anwendung im Bereich der Virtuellen Konstruktion, des CODY-Virtuellen Konstrukteurs (CVK). Die im Kontext dieser Arbeit angesprochenen Teilgebiete werden in Abbildung 1.4 schematisch verdeutlicht.

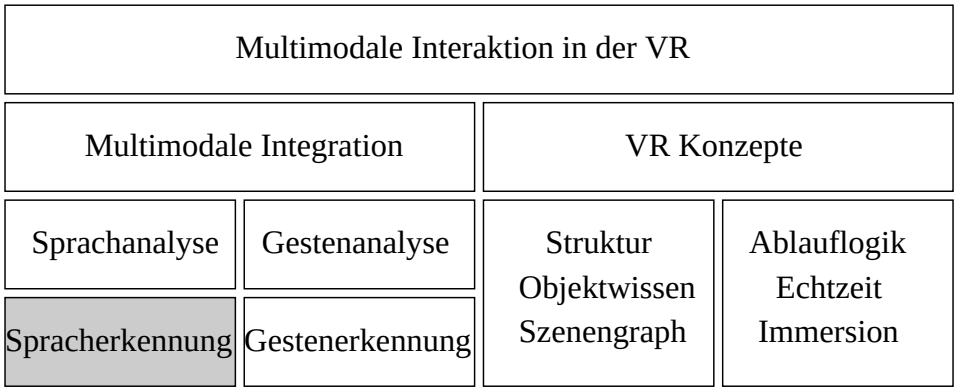

Abbildung 1.4: Thematisches Spektrum der betrachteten Gebiete (mit Fokussierung auf die hell dargestellten Teilgebiete).

Als Grundlage der gestellten Aufgabe dient eine Aufstellung der umzusetzenden Interaktionen in dem Bereich der VR/VK. Weiterhin müssen diejenigen gestischen und sprachlichen Anweisungen identifiziert werden, welche im Kontext dieser benötigten Interaktionen a) vom kommunikativen Aspekt her adäquat erscheinen und daher b) in dem gewählten Szenario mit hoher Wahrscheinlichkeit geäußert werden. Diese Äußerungen dienen anschließend als aktionsauslösende und interaktionsvermittelnde Einheiten.

Um eine kombinierte Auswertung von Gestik und Sprache vorzunehmen, wird eine gemeinsame Integrationsbasis benötigt. Dabei gibt es zwei verschiedene Ansätze: Entweder werden präsemantische oder semantische — auf Bedeutung basierende — Einheiten in beiden Modalitäten betrachtet. Zur Entwicklung einer dementsprechenden Methodik und der anschließenden Konzeption eines Integrationssystems müssen folgende Fragestellungen bearbeitet werden:

- Welches sind die Integrationsziele?

- Welches sind die verschiedenen modalen integrierbaren Einheiten?
- Welche dieser Einheiten sind komplementär oder redundant?
- Wie sind also die phänomenologischen und semantischen Zusammenhänge definiert?

Der Beantwortung dieser Fragen wird dabei auf allen Ebenen, vom Konzept über das Modell bis hin zu einer technischen Realisierung nachgegangen. Dabei wird weniger auf den Bereich der eigentlichen Spracherkennung, dem Weg vom Sprachsignal zum identifizierten Wort eingegangen (s. graue Markierung in Abbildung 1.4). Dieser bildet ein eigenständiges Forschungsgebiet und wird u.a. in einem Kooperationsprojekt bearbeitet [24]. Es besteht im Kontext der Sprachverarbeitung allerdings der Bedarf, die im Bereich des NLP (Natural Language Processing) angewandten Methoden auf mögliche Erweiterungen zur Interpretation multimodaler Eingaben hin zu analysieren, um diese Konzepte gegebenenfalls zu adaptieren und zu erweitern. Dagegen muss als Basis einer Auswertung multimodaler Eingaben ein Verfahren entwickelt werden, um die eigentliche Gestenerkennung zu realisieren. Auf diesem Gebiet gibt es keine etablierten Methoden; es befindet sich — verglichen mit dem Forschungsstand in der Spracherkennung — in einem recht frühen Entwicklungsstadium.

Resultierend sollen die Ergebnisse in der Ausarbeitung adäquater Verfahren für einen Einsatz in VR-Umgebungen münden. Die Entwicklung von Ansätzen, zur Integration solcher — im Datenfluss und Auswertungsprozess sehr unterschiedlicher — Konzepte im Kontext einer Anwendung der Virtuellen Realität, ist ein Schwerpunkt dieser Arbeit. Der aus der Computergraphik entstandene junge VR-Forschungzweig hat, gegeben durch die Anforderungen der Realitätsnähe und der damit ebenfalls geltenden Echtzeitbedingungen, ein breites Spektrum spezieller eigener Verfahren und Techniken entwickelt. Die in den letzten Jahren begonnenen Bestrebungen versuchen hier einheitliche Konzepte zur Daten- und Prozessmodellierung für VR-Entwicklungen bereitzustellen, um z.B. eine weitgehende Flexibilität in der Konfiguration vorhandener Ausgabegeräte zu ermöglichen. Im Zusammenhang dieser Arbeit wird dagegen eine Konzeption zur Bereitstellung einer modularen und transparenten multimodalen Eingabe- und Interaktionsmöglichkeit entwickelt. Der Fokus liegt nicht auf der Realisierung möglichst naturgetreuer Ausgabemöglichkeiten, sondern in der Ermöglichung einer natürlichen Kommunikation in und *mit* Virtuellen Umgebungen sowie einer modularen Einbettung der dazu entworfenen Verfahren in aktuell bestehende und neu entwickelte VR-Modellierungskonzepte.

Die Teilziele können weiter zusammengefasst werden. So lassen sich die bisher erläuterten Aufgabenstellungen wie folgt konkretisieren:

- Spezifikation benötigter und geeigneter Interaktionen
- Realisation gestischer Eingabe und deren Analyse
- Identifikation der multimodalen Integrationsbasis
- Integration und Interpretation der multimodalen Äußerungen
- Einbettung in eine VR-Anwendung (z.Bsp. Virtuelle Konstruktion)
- Entwicklung von Konzepten zur Verankerung der Interaktion in Szenengraph- und Ablaufstrukturen

Zu diesen Zielen werden in den folgenden Kapiteln unter den jeweiligen geltenden Rahmenbedingungen der Teilaufgaben möglichst weitgehend generelle Lösungsansätze unter Berücksichtigung bisheriger Forschungsergebnisse entwickelt. Diese werden anschließend im praktischen Einsatz anhand einer exemplarischen Implementierung eines technischen Systems validiert. Angestrebt werden nicht spezielle einmalige Lösungen, sondern zum einen die Bereitstellung von Verfahren zur multimodalen Integration unter Berücksichtigung eines kognitiv angelehnten methodischen *Groundings* auf inspektiv und experimentell erhaltener Informationen über sprach-gestische Kommunikation. Zum anderen sollen geltende Quasistandards bei der Modellierung von Systemen der Virtuellen Realität methodisch um neuartige Interaktionsschnittstellen erweitert werden.

1.3 Überblick

Nach der hier erfolgten Motivation und Definition der Zielsetzung wird in Kapitel 2 eine Aufarbeitung des Forschungstandes in drei thematisch unterschiedliche Bereiche gegliedert. Abschnitt 2.1 liefert eine Übersicht bisheriger Ansätze, um multimodale Eingaben sowohl in computergraphischen als auch in VR-basierten Anwendungen nutzbar zu machen. In Abschnitt 2.2 wird auf die Modalitäten Gestik und Sprache eingegangen und eine Definition der Verwendung findenden Begriffe vorgenommen. Betrachtungen über Gestik in Bezug auf Phänomenologie und kommunikativen Gebrauch sowie über Beobachtungen und Modelle zu multimodalen — sprach-gestischen — Zusammenhängen, münden in einem Vergleich der konzeptuellen Möglichkeiten einer Gestenerkennung und -analyse mit bestehenden Methoden der Spracherkennung. Abschnitt 2.3 dagegen konzentriert sich auf die Domäne der Virtuellen Realität. Es werden die primären Eigenschaften des *Virtuellen* in den VR-Umgebungen erläutert sowie die dort existierenden Eingabemethoden diskutiert, um den Bedarf an neuen Eingabemetaphern zu veranschaulichen. Abschnitt 2.3.1 konkretisiert dieses anhand der domänenspezifischen Anforderungen einer speziellen VR-Anwendung, eines Systems zur Virtuellen Konstruktion und den darin benötigten Interaktionsoperationen. In der sich anschließenden Diskussion resultieren diese Betrachtungen in der Einführung multimodaler, sprach-gestischer Interaktionen als neuartige Schnittstelle für VR-Umgebungen, sowie in der Identifikation sowohl spezifischer für diese Aufgabe nützlicher Gestenklassen, als auch in der Spezifikation von Anforderungen an eine Auswertungsmethodik.

Im Kapitel 3 erfolgt eine genaue Betrachtung von möglichen Gestenformen um eine multimodale Interaktion in der VR zu gewährleisten. Nach der generellen Unterteilung möglicher Interaktionsabläufe in zwei verschiedene Typen, werden hier abschließend die zu bewältigenden Teilaufgaben spezifiziert.

Mit einer Betrachtung möglicher Gestenmerkmale beginnt der erste große Abschnitt in Kapitel 4. Abschnitt 4.2 führt die zentralen Konzepte der Aktuatoren und der Attributsequenzen als Ausgangspunkte für ein Gestenerkennungssystem in der VR ein. Nach der Ausarbeitung verschiedener Aktuatortypen widmet sich — auf den Aktuatoren aufbauend — der folgende Teil (Abschnitt 4.3) der eigentlichen Gestenerkennungsaufgabe. Anhand von speziellen Testprädikaten wird ein Templatematchingverfahren für die Erkennung und Auswertung von bezüglich der Interaktionsgestaltung nützlichen und wesentlichen Gesten erarbeitet. Am Schluss dieses Kapitels werden in Abschnitt 4.4 vorhandene VR-Interaktionswerkzeuge um eine Art Adap-

terkonzept erweitert. Sogenannte Motion-Modifikatoren erlauben es, mittels gestischer Äußerungen interaktive Szenenveränderungen vorzunehmen. Diese Verfahren dienen später der schritthaltenden Interaktion mit einem VR-System.

Kapitel 5 befasst sich mit der Aufgabe der multimodalen Integration und Interaktionssteuerung. Basierend auf einer Untersuchung und Analyse möglicher verbaler Äußerungen in Bezug auf die umzusetzenden Interaktionen werden im ersten Teil die korrespondierenden Einheiten im Sprach- und Gestenkanal herausgearbeitet. Als Resultat einer Betrachtung in Frage kommender Ansätze für die Integrationsaufgabe fokussiert Abschnitt 5.3 auf die Erweiterung des ATN-Formalismus für die gemeinsame Auswertung sowohl von sprachlichen als auch von gestischen Perzepten. In diesem erweiterten Modell können neben temporalen und semantischen Zusammenhängen ebenso evtl. geltende Anwendungzustände in einer gemeinsamen Repräsentation ausgedrückt werden. Dieses Modell erlaubt weiterhin die Beschreibung funktionaler Operationen und bietet damit die Voraussetzungen, um die Interpretation und Auswertung multimodaler Äußerungen einzubinden. Darauf aufbauend schließt das Kapitel mit der Ausarbeitung von für die multimodale Interaktion besonders wichtigen Auswertungsvorgängen.

Die Dokumentation einer exemplarischen technischen Umsetzung der erarbeiteten Konzepte erfolgt in Kapitel 6. Beispielinteraktionen aus dem SGIM-System skizzieren bildlich eine Auswahl aus den hier als besonders nützlich betrachteten Interaktionsoperationen. Das anschließende Kapitel zieht ein Resumee über die in dieser Arbeit entwickelten Konzepte und diskutiert die Ergebnisse. Abschließend wird ein Ausblick gegeben. In diesem werden sowohl Modifikationen und Erweiterungen der entwickelten Methoden, als auch zukünftige Möglichkeiten, die hier eingeschlagene Forschungsrichtung fortzusetzen, betrachtet.

Kapitel 2

Stand der Forschung

Als Grundlage einer Betrachtung der im Kontext von multimodaler Interaktion in Virtuellen Umgebungen relevanten Arbeiten erfolgt zuerst eine Aufarbeitung des Forschungsstandes aus den dafür wichtigen Gebieten. Diese können thematisch generell in drei unterschiedliche Bereiche gegliedert werden:

1. Multimodale Interaktionssysteme in der Computergraphik haben schon recht früh, begonnen am Anfang der 80er Jahre, das Forschungsinteresse geweckt, da diese Systeme — zentriert auf eine graphische Repräsentation — eine grundlegend andere Bedienung als herkömmliche, auf Textbildschirmen basierte Terminals nahelegen. Seit diesen Anfängen gab es daher immer wieder unterschiedliche Bestrebungen neuartige Interaktionsformen mit Hilfe der natürlichen Modalitäten Gestik und Sprache zu realisieren. Die in Abschnitt 2.1 folgende Betrachtung realisierter Systeme soll eine Übersicht in Anwendungsgebiete, Interaktionsformen und methodische Grundlagen geben.

2. Gestik und Sprache sind unsere primären Kommunikationsmittel. Im Gegensatz zur Gestik, welche teilweise als prä- oder sogar alinguistische Kommunikationsform gesehen wurde, haben sich für die Erforschung der Sprache eigene Forschungsgebiete etabliert. Abschnitt 2.2 stellt die für diese Arbeit relevanten Ergebnisse zur Phänomenologie und Semantik menschlicher Gestik zusammen und erläutert die zu berücksichtigenden Ergebnisse zur Korrespondenz sprachlicher und gestischer Äußerungen. Darauf basierend werden technische Verfahren sowohl zur Gestenerkennung und -analyse, als auch zur multimodalen Integration betrachtet.

3. Virtuelle Realität und konkret eine Anwendung der Virtuellen Konstruktion sind die Kernthemen in Abschnitt 2.3. Sie repräsentieren das hier gewählte Einsatzgebiet für eine multimodale Interaktion. Die in diesem Forschungsgebiet etablierten Methoden stellen sehr spezifische Anforderungen vor allem an die Echtzeitfähigkeit und die Einhaltung möglicher Latenzzeiten der Operationen. Als Grundlage einer Anwendungs- und Interaktionsmodellierung haben sich in der VR Organisationsstrukturen für benötigte graphische und logische Entitäten herausgebildet. Diese gewährleisten eine portable Beschreibung a) der virtuellen Szene, b) von möglichen (bisher WIMP-basierten) Interaktionen mit dieser und c) von möglichen Ausgabegerätekonfigurationen.

2.1 Multimodale Eingabesysteme in der Computergraphik

Die bisherigen Vorhaben, multimodale Interaktionsmetaphern für Computergraphik (CG) oder VR-gestützte Computeranwendungen und Umgebungen zu schaffen, unterscheiden sich erheblich im Grad und in der Art der ausgewerteten Modalitäten. Im Gegensatz zur Spracherkennung umfasst der Begriff der Gestenerkennung ein sehr heterogenes Forschungsfeld. Dieses beinhaltet beispielsweise neben der Interpretation mit einem Stift gezeichneter Eingaben ebenso die Erkennung der Symbole von Gehörlosensprachen als auch eine Analyse *natürlich* geäußerter sprachbegleitender Körperbewegungen.

Die VR hat sich erst langsam als eigenständiges Forschungsgebiet aus der Computergraphik entwickelt (s. dazu auch Abschnitt 2.3). Parallel zu dieser Entwicklung finden wir im Bereich der MMK verschiedene Anwendungsgebiete einer multimodalen Interaktion, angefangen bei der Manipulation einfacher 2D-Darstellungen bis hin zu Versuchen der Steuerung immersiver VR-Systeme in den 90er Jahren. Besonders letztere setzten eine grundsätzlich andere konzeptuelle Vorgehensweise bei der Modellierung und der Verarbeitung multimodaler Eingaben unter den Randbedingungen Interaktion und Immersion voraus — ein Umstand, welchem viele der betrachteten Arbeiten kaum Beachtung geschenkt haben, sodass auf diesem Gebiet keine gültigen Lösungsansätze verfügbar sind.

In diesem Abschnitt werden bisherige Bestrebungen der Exploration von auf menschlicher Gestik und Sprache basierender MMK-Methoden unter verschiedenen Gesichtspunkten betrachtet. Besondere Beachtung soll den in den verschiedenen Ansätzen zur Eingabesteuerung berücksichtigten Typen und

Formen von Gestik im sprachlichen Kontext geschenkt werden. Weiterhin sind die Art der gesteuerten Anwendung im Hinblick auf die computergraphische Realisierung und die Interaktivität zu unterscheiden: Handelt es sich um eine 2D, 2.5D oder 3D-Präsentation? Ist die Outputgenerierung und/oder Eingabeauswertung interaktiv? Und in welchem Maße ist damit der Grad der Benutzerimmersion ausgeprägt?

Put-That-There [8], ein am MIT (Massachussetts Institute of Technology) entwickeltes System, wird in der Literatur häufig als der Urahn dieser Bestrebungen verstanden. Dieses im Media-Room des MIT verankerte System erlaubte die gemeinsame Benutzung von einfachen sprachlichen Begriffen und einer *Armausrichtung*, um eine Anwendung zu steuern. Als "Gestenerkennung" fungierte die Auswertung der Richtung einer Benutzerextremität (eines Arms). Dieser Arm steuerte — einem Laserpointer gleich — einen Cursor auf einer zweidimensionalen Projektionswand mit statischen Objekten. Wurden nun sprachlich unterspezifizierte Referenzen im Dialog mit dem System geäußert, wie im Beispiel: User: " ... put **that**", System: "`where?`", User:"**there**", diente die Cursorposition der Auswertung und Feststellung unter diesem Cursor befindlicher Objekte und/oder Orte. Das System konnte nach der so erfolgten eindeutigen Identifikation der Referenz nun eine diskrete Zustandsänderung — eine Positionsänderung des Objektes — durchführen. Betrachtet man dieses System unter dem Aspekt der eingesetzten Auswertung der Armrichtung, so ist es zumindest fraglich, ob es sich in diesem Fall um Gestik im eigentlichen Sinne handelt. Ohne Vorgriff auf noch folgende Analysen natürlicher menschlicher Gestik ist dieses zumindest anzuzweifeln, da bei dem beschriebenen Gebrauch des Arms wesentliche gestische Expressionseigenschaften nicht berücksichtigt wurden.

Zum Verständnis erfolgt ein Vorgriff auf Abschnitt 4.1.1: Unter den gestischen Expressionseigenschaften werden dort die während der Ausführung der Geste beobachtbaren Formmerkmale bezeichnet. Als solche werden die eingenommenen Positionen und Ausrichtungen der die Geste ausführenden Körperextremitäten sowie die besonderen dynamischen Eigenschaften — etwa Ruhephasen oder Beschleunigungen — während der gestischen Ausführung verstanden. Die zeitliche Verortung verschiedener spezifischer Expressionsmerkmale zeichnet eine einzelne Geste für sich aus und macht sie als solche identifizierbar. Diese Merkmale wurden im beschriebenen Put-That-There-System nicht berücksichtigt. Dennoch war dieses System wohl der erste Versuch, zwei natürliche Eingabemodalitäten auszuwerten, um damit eine 2D-computergraphische Anwendung zu steuern.

Viele der folgenden, in den 90er Jahren entstandenen Arbeiten konzentrieren sich besonders auf den schon kurz angedeuteten Vorgang der Verknüpfung unterschiedlicher Eingabequellen [32][45][63][55]. Untersucht wurden hier vor allem Möglichkeiten, die Benutzerdeixis — in Bezug auf das gestische Bedeuten als auch auf das sprachliche Benennen computergraphisch dargestellter Objekte mit Hilfe von Text- oder Spracheingaben — nutzbar zu machen, wobei der Schwerpunkt auf der Umsetzung der multimodalen Integration lag. Als Anwendungsdomäne findet sich selbst bei neueren Arbeiten [3] — wie bereits durch Put-That-There initiiert — häufig nach wie vor die Bedienung von Systemen mit 2D-basierter Oberfläche.

Im Kontext von speziell in der VR verankerten Arbeiten findet sich im **GIVEN** Projekt [6][7] ein Ansatz, gestische Eingaben in einer VR-Anwendung einzusetzen. In diesem Projekt dienen symbolische Gesten (s. die dazu noch folgende Übersicht in Tabelle 2.1) als ein Teilbereich menschlicher Gestik, der Steuerung. Ein erkanntes gestisches Symbol — zum Beispiel ein nach oben gerichteter Daumen — löst eine spezifische Operation aus. Als Grundlage dieser Vorgehensweise findet eine einfache Abbildung einer Menge bedeutungsvoller Handformen auf eine Menge diskreter Anwendungskommandos statt. Durch die dadurch gegebene Unmissverständlichkeit in der Interpretation und die Eindeutigkeit dieser in ihrer Form meist sehr unterschiedlichen Gesten, findet man ein solches Vorgehen in vielen einfachen VR-Schnittstellen mit Gesteneingaben. Die Auswertung dieser symbolischen Gesten wird durch die im VR-Umfeld sehr häufig vorhandene Sensorik, die Verfügbarkeit von Datenhandschuhen und Positionssensoren, begünstigt. Erste Versuche mehr als rein symbolische Gesten mit einer Spracherkennung zu koppeln, um eine virtuelle Szene zu manipulieren, finden sich bei Weimer und Ganapathy [98]. Hier werden vor allem translatorische Aspekte der Armgestik ausgewertet, um graphisch dargestellte Kurven zu beschreiben und zu modifizieren.

ICONIC: Der Ansatz der *Dereferenzierung*[1] von Objekten und deren Ortsänderungen in einer 3D-Szene mit Hilfe von Spracheingabe und gestischer Formbeschreibung durch *depictive gestures* wird von Sparrell und Koons im ICONIC System [87][45] exploriert. Wichtige Formmerkmale der Objekte, die Position von Ecken, die Lage von begrenzenden Flächen und Hauptachsen und eventuelle intrinsische Vorder- und Oberseiten dienen als Repräsentation der Objekt-Formbeschreibung. Ein mehrstufiges merkmalsbasiertes Ge-

[1] Dereferenzierung bezeichnet im Folgenden den Vorgang der Bestimmung des oder der Referenten.

stenanalysesystem liefert aus den Sensorrohdaten von Positionssensoren und Handschuhen diskrete Gestensegmente. Betrachtet werden die Fingerbeugungen und die in sechs Richtungen erfolgenden Bewegungen zweier *virtuell* in der Hand lokalisierter Achsen: Eine aus der Handinnnenfläche zeigende und einer in Richtung der Fingerorientierung liegende Achse. Zwischen diesen beiden Repräsentationen wird ein *iconic mapping* gesucht, eine Übereinstimmung in den erkannten Gestenmerkmalen und den Objektmerkmalen. Sprache und Gestik werden anschließend über einen Frameformalismus integriert und die 3D-Szene daraufhin gemäß der Interaktion verändert. Besondere Beachtung erfährt eine erweiterte Version eines ebenfalls merkmalsbasierten Gestenerkenners bei Wexelblatt [100][101], Arbeiten, welche ebenfalls am MIT stattfanden und im gleichen Kontext gesehen werden können. Wexelblatt liefert auch einen guten Überblick des betrachteten Gebiets und beurteilt bisher stattgefundene Versuche der Interaktion mit symbolischen Gesten ebenfalls kritisch, betrachtet bei seinen Arbeiten selber aber primär das entwickelte System zur Gestenerkennung, also rein unimodale Eingaben.

Cavazza et al. [17] greifen den ursprünglichen Gedanken des deiktischen Dereferenzierens durch eine Kombination von Zeigegesten (pointing) und Spracheingabe wieder auf. Ziel der Arbeiten ist der explizite Einsatz multimodaler Eingaben in Virtuellen Umgebungen. Kritisch werden die dort sehr häufig realisierten Benutzerschnittstellen, beispielsweise die Benutzung von 3D-Menüs, aufgrund der Unnatürlichkeit ihrer Benutzung in den Virtuellen Räumen betrachtet. Sie erweitern den Begriff des *pointing* in ihrem Umfeld zum *extended pointing*. Damit wird ein Bestimmungs- oder Benennungsproblem bezeichnet, in welchem die Gestik als mehr als ein 3D-Mausklick Verwendung findet. Das Ziel der *extended pointing*-Gesten ist die Bestimmung des oder der Referenten. Analytisch betrachtet können also genauso eine Entität (Objektelement oder Richtung im virtuellen Raum) oder mehrere durch den bezeigten Raum gruppierte Objekte gemeint sein.

Ein auf dem TAG (Tree-Adjoining Grammar) Formalismus operierendes NLP (Natural Language Processing) Modul analysiert hier die Spracheingabe des Benutzers und holt sich bei Bedarf komplementäre Informationen aus dem Gestenmodul. Für die Gestenerkennung wird ebenfalls der schon erläuterte, am MIT entwickelte Gestenparser eingesetzt. Zwei exemplarische VR-Anwendungen dienen der Erprobung der Auswertung deiktischer Gesten: Telerobotik und VE-Dokumentation. Weiterhin wird die in den aktuellen Arbeiten nicht betrachtete aber als nützlich erachtete Auswertung der Blickrichtung zur Dereferenzierung betont sowie für VR-Anwendungen der Bedarf an echtzeitfähigen Verfahren zur Gestenanalyse herausgestellt.

VisSpace: Zu den aktuellen Arbeiten im Bereich multimodaler Schnittstellen für computergraphische Anwendungen zählt das von Lucente et al. [57] am IBM Research Center in New York entwickelte VisSpace oder Dreamspace Projekt. Bemerkenswert ist, dass hier weitgehend auf am Benutzer befestigte Sensoren verzichtet wird. Ein Anwender steht in einem ca. 4x8 m großen Raum vor einem Großbildschirm und kann mit Hilfe von kontinuierlicher Sprach- und Gesteneingabe die dargestellte Szene manipulieren. Eine am MIT Projekt *Pfinder* (PersonFinder) angelehnte Software ermittelt über eine am oberen Rand der Projektionswand befestigte Kamera die Positionen von Kopf, Händen, Füßen, Ober- und Unterkörper mit einer Rate von ca. 10Hz. Die Spracheingabe erfolgt über ein kleines Kragenmikrophon. Interaktionen umfassen die Auswahl von Objekten durch Zeigen und Benennen, das Verschieben in Form eines *Dragging* sowie zweihändige Größenveränderungen ("*make it this big*"), indem während der sprachlichen Äußerung die Hände im gewünschten Abstand voneinander gehalten werden. Besonders letztere, zu den ikonischen Gesten zählende Gestik, wird als nützlich für die Interaktion betont. Auf eine genauere Analyse der Gestik, beispielsweise die einzelnen Handformen beim Zeigen, wird aber verzichtet. Um eine Zeigerichtung zu bestimmen, wird über den Kopf-Hand-Differenzvektor — nach anfänglichem vierfachen Kalibrieren — die Richtung extrapoliert. Die multimodale Integration wird hier durch die Spracheingabe initiiert, anschließend werden benötigte Informationen aus der Gestenanalyse ermittelt. Die Visualisierung der Szene erfolgt bereits in Anlehnung an VR-Verfahren: Unter Verzicht auf eine Stereoprojektion erfolgt eine Aktualisierung der Betrachtungsperspektive gemäß der Benutzerkopf-Position.

Von Virtuellen Agenten zur Kunst Neben den zuvor betrachteten Projekten und Methoden, die durch ihre Art der eingesetzten gestischen und multimodalen Interaktion für diese Arbeit von Bedeutung sind, findet sich eine weitere Übersicht von Systemen zur multimodalen Steuerung bei Thorisson [90]. Diese Arbeit verfolgt dabei einen Ansatz ähnlich dem in Maes et al. [59] [60][61] und Cassell [16]. Gestik und Sprache dienen hier der Interaktion mit einem *Virtuellen Agenten*, einem teil-autonomen Kommunikationspartner in der virtuellen Szene, weniger der direkten eigenen Einflussnahme eines Anwenders auf in der Szene graphisch repräsentierte Objekte. Hier wird besonders der gestische Ausdruck des Benutzers als Eingabemedium ausgewertet, indem der Virtuelle Agent auf die Bewegungen des Benutzers in komplexer Weise reagiert. Dabei ist die gezeigte Reaktion von mehr als der reinen Benutzereingabe abhängig. So spielen intern modellierte Vorgän-

ge über den Gemütszustand und die Wünsche des Agenten, ebenso wie der Kontext der Äußerung eine weitere Rolle bei der Festlegung des Agentenverhaltens. In noch abstrakterer Weise bedienen sich künstlerische Arbeiten der Auswertung gestischer und multimodaler Eingaben. In [47] und [13] werden solche exemplarischen Anwendungen vorgestellt. Hier gibt es aber keine direkte Auswertung der Eingaben auf ihren manipulativen vom Anwender intendierten Gehalt hin, daher können für das Vorhaben der Interaktion mit Virtueller Konstruktion nur wenige Erfahrungen und Ergebnisse direkt Verwendung finden.

Das umschriebene Themenspektrum setzt sich, wie den Beispielen zu entnehmen ist, aus sehr unterschiedlichen Facetten zusammen. Neben den hier besonders erläuterten Arbeiten, in denen Gestik und Sprache zum Zweck einer Manipulation graphisch repräsentierter Objekte eingesetzt wird, finden sich weitere Ansätze multimodaler Kommunikation mit Virtuellen Agenten oder als künstlerische Ausdrucksform. Gemeinsam ist diesen Vorhaben der Ansatz, die natürlichen Modalitäten Sprache und Gestik als Eingabemittel verfügbar machen zu wollen. Hat dabei die Erforschung der menschlichen Sprache in der Linguistik bereits ein eigenständiges Forschungsgebiet mit Methodiken und festen Begriffsbedeutungen, auf welche zurückgegriffen werden kann, kann dieses von der Gestik nicht bestätigt werden. Hier finden sich sehr unterschiedliche Interpretationen und Ansätze. Schon der Begriff der "Gestik" wird nicht eindeutig definiert, die Verfahren diese Gestik zu erkennen und zu analysieren sind dabei sehr vielfältig und unterschiedlich. Daneben finden sich keine generellen Ansätze der Interaktionsmethoden und der Kopplung der beschriebenen Erkenner- und Eingabesysteme mit einer VR-Anwendung. Die folgenden Abschnitte gehen auf diese beschriebenen Themen weiter ein. Neben einer Begriffsklärung erfolgt eine Betrachtung phänomenologischer und kommunikativer Eigenschaften menschlicher Gestik. Daraufhin werden bisherige Forschungsergebnisse zur Multimodalität, speziell zum Zusammenhang zwischen gestischen und sprachlichen Äußerungen aufgearbeitet sowie maschinelle Erkennungsverfahren für Gestik und Sprache verglichen. Abschnitt 2.3 befasst sich anschließend mit dem Begriff der Virtuellen Realität und dem Stand der Forschung in dieser Domäne.

2.2 Gestik und Sprache

Vor einer Betrachtung multimodaler Zusammenhänge, hier immer auf das simultane Kommunizieren mit Gestik und Sprache bezogen, wird im folgenden Abschnitt mit einer phänomenologischen Betrachtung menschlicher gestischer Äußerung an Hand eines Modells des menschlichen Bewegungsapparates begonnen. Insbesondere sollen die verwendeten Begriffe Geste, Postur und Mimik für den weiteren Gebrauch in der hier benutzten Interpretation definiert und eingegrenzt werden. Grundlage für die folgenden Ausführungen sei: a) ein kinematisches Modell des biologischen Aufbaus des menschlichen Bewegungsapparates und b) dessen Verwendung unter dem Aspekt der Kommunikation, also der Weitergabe von Informationen im weitesten Sinn. Es soll hier keine Aussage über den allgemeinen Diskussionspunkt, ob nicht jedwede Bewegung — da durch sie Information übermittelt wird — Kommunikation darstellt, getroffen werden. Es wird stattdessen insbesondere unter dem Aspekt des Einsatzes gestischer Eingabe in der MMK der pragmatische Standpunkt einer der gestischen Äußerung zugrundeliegenden kommunikativen Intention eingenommen. Die bewusste Benutzung eines Systems durch einen Anwender impliziert seinen Interaktionswunsch, seine Äußerungen können daraufhin bereits als diesem Zweck dienlich interpretiert werden.

2.2.1 Phänomenologie gestischer Äußerung

In einem kinematischen Modell des menschlichen Bewegungsapparates werden — vereinfacht — starre Verbindungen (**Links**) über Gelenke (**Joints**) mit Hilfe der Muskeln zueinander im jeweiligen Gelenkspielraum bewegt. Diese Art von Änderungen kann prinzipiell in verschiedenen Bezugssystemen modelliert werden. Auf Seite eines (Gesten-)Produzenten werden viele Betrachtungen über die zugrundeliegenden motorischen Fähigkeiten und Kreisläufe im Gelenkwinkelraum dieses kinematischen Modells angestellt. Durch die Veränderung der Parameter in diesem Raum kann die relative Konfiguration der Links zueinander verändert werden und resultierend die absolute Position und Ausrichtung, beispielsweise der einzelnen Gliedmaßen, im euklidischen Raum festgelegt werden. Die Betrachtung der Konfigurationsänderung im euklidischen Raum entspricht dabei, gegeben durch die menschliche visuelle Sensorik und darauf basierender Modelle interner Raumrepräsentation, eher der Sichtweise eines (Gesten-)Rezipienten. Die Abbildung vom Gelenkwinkelraum in den euklidischen Raum wird dabei durch Methoden der direkten Kinematik, die inverse Abbildung respektive durch inverse Ki-

nematik geleistet. Legt man das kinematische Modell als Grundlage der menschlichen Bewegungsausführung zugrunde, lassen sich so Betrachtungen über die allgemeine Bewegungs- und damit auch über die spezielle Gestenphänomenologie in verschiedenen Bezugssystemen anstellen.

Daneben wird durch Muskelbewegung, wie sie für die reale biologische Konfigurationsänderung des menschlichen Körpers ausgeführt wird, die Haut als äußere Hülle des Körpers verformt. Auch diese Veränderung ist durch menschliche Sensorik erfassbar und wird als Merkmal wahrgenommen. Im Zusammenhang dieser Arbeit wird aber besonders eine Untermenge aller möglichen Konfigurationsänderungen betrachtet, menschliche Gestik, also die Bewegungen, welche als *bedeutungstragend* im Sinne einer zwischenmenschlichen Kommunikation identifiziert werden können und welche nicht einem reinen Bewegungszweck dienen. Folgende Arbeitsdefinitionen finden daher Verwendung:

Definition 1 (Geste) *Eine Änderung der Körperhaltung, welche bedeutsame Signale aussendet oder genauer: Eine Geste sei eine dynamische Abfolge von äußerlich sichtbaren Konfigurationsänderungen des menschlichen Bewegungsapparates, welche einer Kommunikation dient.*

Definition 2 (Postur) *Eine Postur sei eine durch eine gesonderte Ruhephase ausgezeichnete Körperkonfiguration in der dynamischen Ausführung einer Geste.*

Definition 3 (Mimik) *Mimik sei die Untermenge der Gestik, welche hauptsächlich durch Tonusänderungen der Muskeln des Gesichtsbereichs im Gesichtsrelief zum Ausdruck kommt.*

Die Definitionen beziehen sich immer auf die durch einen Rezipienten äußerlich erfassbaren Veränderungen des menschlichen Körpers und mithin des Bewegungsapparates. Die folgenden Ausführungen stützen sich bei einer Betrachtung der besonderen Bewegungsmerkmale dabei auf die Sichtweise im euklidischen Raum. Durch die dort resultierenden Bewegungen der Extremitäten erhält man als Bewegungsmerkmale Faktoren, welche durch Gestalt oder Form sowie durch Bewegung, verstanden als dynamische Formänderungen in der Zeit, definiert sind.

Bewegungen eines Körpers im Raum können mittels von diesem Körper durchlaufene *Bahnen* modelliert werden. Wird der Bahnverlauf über

die Zeit betrachtet, erhält man beschriebene *Trajektorien*. Sehr ausführliche Untersuchungen von durch die Extremitäten beschriebenen Trajektorien im Zusammenhang mit menschlicher Gestik findet sich in den Arbeiten von Gaglio [28] und Morasso [69]. Die darin angestellten Untersuchungen veranschaulichen bildlich, dass schon die beschriebenen Bahnen während der Bewegungsausführung gewissen ungewollten Ablenkungen unterliegen. Sollen geometrische Primitive, etwa Kreisabschnitte oder Geraden gestisch dargestellt werden, so unterliegt der menschliche Bewegungsapparat kinematischen und motorischen Beschränkungen, durch welche ein antizipierter Idealverlauf in bestimmten Abschnitten ausgelenkt oder abgefälscht wird. Dennoch fallen diese "Fehler" in der zwischenmenschlichen Kommunikation nicht ins Gewicht, da auf Seite des Rezipienten diese Art von Ungenauigkeiten korrigiert wird. Die während der ausgeführten gestischen Bewegung durch die Extremitäten beschriebenen Bahnen im Raum sind also nur in Maßen präzise, ihre bedeutsamen geometrischen Eigenschaften lassen sich aber aus ihnen abstrahieren.

Gestische Äußerung ist aber nicht allein als ein spatiales Phänomen beschreibbar. Der Begriff der Trajektorien impliziert neben einer rein räumlichen Betrachtung der Bahnen ebenso dynamische Aspekte durch den Bewegungsverlauf in der Zeit. Diese zeitliche Komponente — die temporale Struktur — vergleicht Kendon [42] mit Strukturkomponenten einer simultanen verbalen Äußerung (s. dazu auch Abschnitt 2.2.3). Er identifiziert daraufhin auf unterer Ebene die Organisationsstruktur einer *gesticular phrase* oder *G-phrase*, einer Bewegungssequenz mit den in Abbildung 2.1 dargestellten drei unterschiedlichen Hauptphasen:

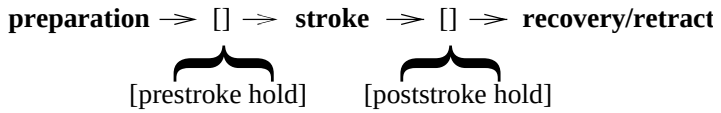

Abbildung 2.1: Phasen einer gestischen Äußerung. Drei Hauptphasen folgen, evtl. unterbrochen durch zwei Holds (Stops), aufeinander.

Die preparation-phase dient dem "Anfahren", also der Vorbereitung des für die gestische Äußerung besonders wichtigen Bewegungsabschnitts, der Strokephase. Dieser *Gestenstroke* entspricht dem — in Bezug auf den Informationsgehalt — wesentlichen Bewegungsabschnitt im spatiotemporalen Verlauf. Der Begriff Stroke ist hier weniger als Schlag oder Streich zu interpretieren, sondern im Zusammenhang der gestischen Bewegung als Gesten*strich*. Bei der damit beschriebenen Phase handelt es sich nicht nur um

einen diskreten Zeit*punkt*, sondern durchaus auch um eine Zeit*spanne* kontinuierlicher Bewegung. In der retract-phase wird die Geste anschließend entweder bis in eine Ruheposition abgebaut oder in eine folgende gestische Äußerung übergeleitet.

Mehrere G-phrases konstituieren sogenannte *gesticular units* (*G-units*). Diese zeichnen sich im Gegensatz zu den G-phrases durch die Rückkehr in eine absolute Ruheposition aus. Zwischen G-Phrases kommt es zu Übergängen, in welchen eine solche absolute Ruheposition nicht eingenommen werden muss.

Durch diese temporalen Faktoren werden auf einer höheren Ebene neben den rein spatialen ebenso dynamische Merkmale geäußert. In [43] beschreibt Kendon diese Merkmale als Bewegungscharakteristiken, welche es einem Rezipienten erlauben, intentionale gestische Äußerungen als *Figur* oder *Gestalt* vor einem Bewegungshintergrund zu erkennen. Er identifiziert sie als bezeichnend für offensichtlich beabsichtigten gestischen Ausdruck. Die durch Morasso [69] analytisch gewonnenen Erkenntnisse bestätigen das Vorhandensein dieser Merkmale für eine Auswahl von gestischen Bewegungen als bedeutsame Muster sowohl im Gelenkwinkel- als auch im euklidischen Raum.

Die Gesamtheit sowohl der spatialen als auch der durch die Dynamik gegebenen temporalen Merkmale werden hier im weiteren als *spatiotemporale Expression* der Gestik bezeichnet. Gestische Äußerungen bilden somit *Muster* der über die Zeitdauer der Äußerung emittierten Merkmale aus. Diese Muster sowie die Ausdrucksstärke der einzelnen Merkmale darin ermöglichen es, gestischen Ausdruck von rein operationalen, praktischen oder natürlichen Bewegungen zu trennen.

Definition 4 (Spatiotemporale Gestenexpression) *Die spatiotemporale Gestenexpression sei die Gesamtheit der signifikanten Merkmale der die Geste ausführenden Extremitäten in Bezug auf: (1) die beschriebene spatiale Bahn, (2) die Beschreibungsdynamik durch die Trajektorie sowie (3) die daraus auf einer höheren Ebene resultierenden Formen (Posturen) im Verlauf der Gestenäußerung.*

Gestik unterscheidet sich hierdurch von gesprochener Sprache, da sich durch diese Expressionseigenschaften ein weites Potential möglicher Modifikationen einer gestischen Äußerung ergibt. In der Sprache werden unterschiedliche Begriffe und Konstruktionen verwendet, um die zu übermittelnde Information zu kommunizieren. Stehen diese Begriffe und Strukturen einer verbalen Äußerung fest, sind Modulationen nur begrenzt in der Intonation und der Sprechgeschwindigkeit möglich. Gestik dagegen kann in Bezug auf

den gleichen zu kommunizierenden Inhalt sehr unterschiedlich ausgeprägt ausgeführt werden und sowohl in den Merkmalsmustern als auch in der Ausdrucksstärke variieren. Bewegungen können darüber hinaus auf einer weiten Skala entweder rein praktischer Natur, Ausdruck von internen Gefühlszuständen oder von intentionalem kommunikativen Verwendungszweck sein. Die Grenzen zwischen diesen Bereichen sind deutlich unschärfer als es bei einer sprachlichen Äußerung der Fall ist.

2.2.2 Gestenklassifikationsschemata

Angefangen in den 40er Jahren mit eher semiotischen Betrachtungen von Efron [22], entstanden viele neuere Arbeiten zur menschlichen Gestik und deren kommunikativen Eigenschaften in den 80er und 90er Jahren. Von Kendon [42] [44] über Nespopulous und Lecours [74] zu McNeill [65] finden wir neben Untersuchungen zur Gestensemantik ebenfalls Betrachtungen zu motorischen Eigenschaften der Gestenexpression oder zu semantischen Bezügen basierend auf temporalen Zusammenhängen zwischen Gestik und Sprache. Gemäß dem unterschiedlichen Betrachterfokus wurden so auch verschiedene Klassifikationsschemata der menschlichen, hier vor allem der *Handgestik* vorgeschlagen.

Um zu einer übersichtlichen Darstellung von Formen dieser gestischen Expression in Bezug zu ihrem kommunikativen Gehalt zu gelangen, wird in Tabelle 2.1 nach [52] der Versuch unternommen, die verschiedenen Resultate basierend auf Arbeiten von Efron [22], Ekman und Friesen [23], sowie Nespoulous und Lecours [74] und ähnlichen zusammenführenden Klassifikationsversuchen in Rimé et al. [81] und Wexelblat [100] gegliedert nach ihrem Kommunikationsaspekt zusammenzufassen. Entscheidend dafür sind nicht Kriterien wie die Art oder Ausprägung einer Geste, ihr temporaler Verlauf oder ob sie kulturell erlernt und einzugrenzen ist, sondern allein die Rolle, welche die Gestik in einem kommunikativen Akt spielt. Unter einem solchen verstehen wir vereinfacht den Vorgang des Austauschs von Informationen zwischen zwei Individuen, welcher einem wohldefinierten Zweck dient. Unterschiedliche Kommunikationsakte lassen sich nach ihrem Zweck unterscheiden, also danach, welche Intentionen hinter einem solchen Akt stehen. Es soll daher an dieser Stelle auf die Vermittlerrolle hingewiesen werden, welche ein zu entwickelndes technisches System einnehmen soll. Dieses soll dabei sowohl die Aufgabe eines Mediators als z.Bsp. in der speziellen Anwendungsdomäne die des Konstrukteurs in einer Instrukteur/Konstrukteur Beziehung erfüllen. Ohne Beschränkung der Allgemeinheit wird sich folglich der Hauptbestandteil der benutzten Kommunikationsakte stark einschrän-

ken lassen. Es wird sich zum überwiegenden Teil um Anweisungen (z.Bsp. des Instrukteurs) handeln, in welchen die gewünschte Veränderung der dargestellten Szene spezifiziert und/oder moduliert wird. Die folgende Tabelle beschreibt daher die verschiedene Gestentypen mit ihren in der Literatur zu findenden Bezeichnungen zusammen mit Beispielen ihres kommunikativen Gebrauchs. Anhand dessen lässt sich definieren, welche Gestentypen später nützliche Interaktionsinformationen für den gewählten Einsatzzweck in einer multimodalen Steuerung liefern können.

Typ	Klassifikationsname	Charakteristika
I	1. Deiktisch	Referenzieren auf Objekt(e), Ort(e) und Richtung(en) im Raum. Bsp.: Jemand zeigt auf einen Stuhl und sagt: „Hol bitte diesen Stuhl ... ", zeigt dann auf eine freie Stelle neben dem Tisch und sagt: „ ... und stelle ihn da hin!".
II	1. Mimetisch 2. Ikonisch 3. Objektbezogen 4. Piktographisch	Die Extremitäten werden als Platzhalter benutzt, um das Verhalten eines beschriebenen Objektes oder Zustandes nachzubilden. Bsp.: Eine sich öffnende Hand demonstriert das Öffnen einer Blüte.
III	1. Physiographisch 2. Kinetographisch 3. Pantomimisch	Repräsentieren und verbildlichen das Zusammenspiel mit einem Objekt. Zeigen die Interaktion bei der Benutzung. Bsp.: Jemand demonstriert die Benutzung von Hammer und Nagel ohne Hilfsmittel.
IV	1. Symbolisch 2. Moduseinstellend 3. Emblematisch	Haben eine eindeutige Semantik als alleinstehende Geste und verändern ggfs. den Modus in welchem eine gleichzeitige verbale Äußerung interpretiert wird. Bsp.: 1. Zeigefinger und Daumen bilden einen Kreis und symbolisieren OK. 2. Die Handfläche zeigt parallel zum Boden und wackelt um die Längsachse als Zeichen, dass eine simultane Aussage unsicher ist.
V	1. Ideographisch 2. Metaphorisch 3. Ikonisch	Veranschaulichen eine räumliche metaphorische Manifestation eines internen Zustands. Beziehen sich auf eine Interpretation. Bsp.: Jemand sagt "mir ist schwindelig" und dreht dabei den Zeigefinger in der Luft.
VI	1. Beats 2. Gestikulation 3. Sprachmarkierend 4. Selbstregulierend	Geben einen Sprachrhythmus an. Betonung und gestische Expression fallen in den gleichen Takt. Bsp.: Redner(in) unterstreicht die entscheidenden Punkte seiner/ihrer Aussage mit einem Klopfen des Zeigefingers auf den Tisch.

Tabelle 2.1: Gestentypen (nach [52])

Illustrative Gesten

Der Ansatz, verschiedene Klassifikationsschemata menschlicher Gestik in einer gemeinsamen Typologie zu vereinen, wird durch die vielfältig unterschiedlichen Betrachtungsebenen erschwert. Die Zuordnung einer Gestenform zu einer gewählten Kategorien ist durchaus nicht eindeutig, wie in Tabelle 2.1 am Beispiel des Begriffs der *ikonischen Gestik* (s. Zuordnung zu Typ II und Typ V) offenbar wird. Als eine in Bezug auf die Exploration multimodaler Interaktion besonders geeignete Einteilung erscheint die von Nespoulous und Lecour [74]. Ihre Definition *Coverbaler Gestik* unterscheidet drei weitere Kategorien: *Illustrative* , *expressive* und *paraverbale* Gesten. Auf die letzteren beiden Kategorien soll hier nicht weiter eingegangen werden, da sie in einem anderen Kontext stehen. Illustrative Gesten dagegen haben starken Bezug zu räumlich manifestierten Eigenschaften. Dieses Merkmal lässt sie als besonders adäquat im Zusammenhang von VR-Interaktionen — wie in Abschnitt 2.3.2 gezeigt wird — erscheinen. Illustrative Gesten unterscheiden sich wiederum in vier Typen, welche folgendermaßen beschrieben werden:

- *Deiktisch:* Das Zeigen auf über lexikalische Einheiten im sprachlichen Kanal geäußerte Referenzen.

- *Spatiographisch:* Das Skizzieren der spatialen Konfiguration des Referenten einer lexikalischen Einheit.

- *Kinemimisch:* Das Beschreiben einer durch eine lexikalische Einheit ausgedrückten Aktionen.

- *Pictomimisch:* Das Beschreiben von Formeigenschaften des Referenten einer lexikalischen Einheit.

Illustrative Gesten betrachten also spatiale Konzepte wie Richtung, Konfiguration, Form und deren zeitliche Änderungen. Diese Faktoren sind besonders für Interaktion mit dreidimensionalen Szenarien wichtig. Die hier erfolgte Einteilung menschlicher Gestik anhand der kommunikativen Eigenschaften dient daher als Ausgangsbasis eines funktionalen Vergleichs und als Abgleich mit den in Abschnitt 2.3.2 noch auszuarbeitenden Interaktionsformen. So wird eine Möglichkeit zur Identifikation von Gestentypen bereitgestellt, welche eine in der VK benötigte Interaktion auf möglichst natürliche Art ausdrücken können.

2.2.3 Korrespondenz multimodaler Äußerungen

Gestik und Sprache werden in der zwischenmenschlichen Kommunikation in der Regel simultan geäußert. Prinzipiell sind so drei mögliche Arten von inhaltlicher Beziehung in sprach-gestischer Information zu unterscheiden:

- Redundanz: Identische Information liegt in beiden Modalitäten vor.
- Komplement: Signifikante Information aus beiden Kanälen ergänzt sich.
- Kontradiktion: Information aus beiden Kanälen steht im Widerspruch.

Eine solche Interpretation setzt eine erfolgte Analyse jeder Modalität für sich voraus. Darauf aufbauend muss eine Zuordnung der Informationen aus beiden Kanälen erfolgen — eine Aufgabe, welche allgemein als *Korrespondenzproblem* [88] beschrieben wird. In erster Linie wird zur Lösung dieses Problems die temporale Beziehung der sprachlichen zu den gestischen Äußerungen betrachtet. Diese drückt implizit die semantische Zusammengehörigkeit zweier multimodaler Expressionen aus, McNeill spricht hier vom semantischen und pragmatischen Synchronismus [65]. Ein temporaler Zusammenhang bedeutet aber durchaus nicht eine einfache Gleichzeitigkeit, sondern kann sehr unterschiedlich und vielschichtig von verschiedenen Faktoren anhängen. Um zu einem Verständnis dieser temporalen Zusammenhänge, und damit zu einer Grundlage und einer Bedarfsspezifikation für ein Konzept eines technischen Integrationsverfahren zu gelangen, werden konkretere Informationen über die relativen zeitlichen Verortungen der multimodalen Äußerungen benötigt. Es wird letztendlich eine Methode gesucht, um beide Modalitäten als Ganzes semantisch zu analysieren.

Die zeitliche Verortung bestimmter Merkmale sprachlicher und gestischer Expression liefert starke Hinweise auf die Zusammenhänge der durch Gestik und Sprache kommunizierten Fakten. Butterworth & Beattie [12], Kendon [42] und später McNeill et al. [65][66] identifizieren beispielsweise eine Kopplung zwischen der Gestenklimax, dem Zeitpunkt der expressivsten Gestenausprägung bei bestimmten Gesten, und der dazugehörigen verbalen Äußerung. Die Geste erfolgt in ihren Betrachtungen im Zeitraum kurz vor bis zum Beginn des verbalen Ausdrucks. Als Maß für die Intervalllänge findet man unterschiedliche Angaben, konkrete Studien bestätigen eine vorliegende Variabilität der Intervalllänge im Zusammenhang mit Faktoren wie zum Beispiel der Wortvertrautheit [71]. Sie liegt aber wohl in der Größenordnung eines Intervalls von bis zu einer Viertelsekunde (250ms) Länge. Dieses quan-

titative Maß wurde in Arbeiten von Lenzmann [55] bereits als Basis für eine technische Integration herangezogen.

Levelt und Richardson [56] dagegen bestätigen in ihren Untersuchungen diesen temporalen "Gestenvorlauf" nicht in so eindeutiger Weise. Ihre Ergebnisse lassen eher auf eine Streuung der beiden Modalitäten um einen gemeinsamen Initiierungszeitpunkt schließen. Sie vergleichen eine enge versus einer losen multimodalen Kopplung. Gemäß der *interaktiven Theorie* wird über den zeitlichen Verlauf eine Synchronisation zwischen den beiden Modalitäten erfolgen und sich ein engeres temporales Muster mit laufend fortschreitender rückgekoppelter Interferenz bilden. Eine solche Interpendenz wurde beispielsweise durch die Arbeiten von de Ruiter [21] für eine Anpassung der Sprache an die gestische Äußerung nachgewiesen. Die *ballistische Theorie* geht dagegen von einer losen Kopplung aus. Nach ihr werden Geste und Sprache zusammen "geplant", sobald das motorische System die Ausführungskontrolle erhält, verlaufen Gesten autark.

McNeill [67] bekräftigt die These eines gemeinsamen Zeitpunktes der Äußerungsgenerierung und konkretisiert diesen im Konzept des *growth-point*, des Zeitpunktes der entscheidenden Ausformung einer mentalen Veränderungen eines Gedankens, der eine multimodale Äußerung generiert. Cummins und Port [20] weisen auf einen weiteren zeitlichen Bezug hin. In ihren Arbeiten untersuchen sie rhythmische Zusammenhänge in der sprach-gestischen Äußerung. Sie betrachten also auf dem grundlegenden zeitlichen Verhalten basierende Ober-Muster. Eine Fragestellung, mit welcher man sich hier konfrontiert sieht, ist das Betrachtungsniveau sprach-gestischer Expressionen: Welches sind die rhythmisch geäußerten "Einheiten" und wie deren Querbezüge? Es finden sich gerade in der Betrachtung dieser Art temporaler Äußerungsstruktur Zusammenhänge zwischen präsemantischen Einheiten. Am Beispiel: Intonation ist fraglos ein sehr abstraktes Kommunikationsmittel, ihre physikalische Ausprägung als Änderung der akustischen Wiedergabeenergie aber ist technisch ohne Interpretation — präsemantisch — zu gewinnen. Ähnliches gilt für gestische Bewegungsmerkmale — extreme Beschleunigungsänderungen der Extremitäten beispielsweise, eine rhythmische Kopplung kann zwischen beiden so schon auf einer vorsemantischen Ebene festgehalten werden.

2.2.4 Verfahren maschineller Spracherkennung

In der Entwicklung der maschinellen Spracherkennung wurden bereits Verfahren entwickelt, um aus einem technisch erfassten analogen Sprachsignal die in diesem Signal transportierten Wörter in ihrer symbolischen Notation

zu ermitteln. Obwohl der Fokus in dieser Arbeit nicht auf der Betrachtung von Spracherkennungsverfahren liegt, soll ein Einblick ein Verständnis für die zugrundeliegenden Problemstellungen und Lösungsansätze geben. Dieser Hintergrund erlaubt im Folgenden Unterschiede und Parallelen zu der Problemstellung in der maschinellen Gestenerkennung herauszustellen. Die hier gemachten Ausführungen sollen daher eine Übersicht über das Gebiet erlauben, sie fußen auf den Ausarbeitungen in [82] und bezogen auf den Einsatz in Virtueller Realität auf einer Einführung in [64]. Die Eigenschaften maschineller Spracherkennungssysteme lassen sich nach [29] in der nachstehenden Tabelle zusammenfassen:

Merkmale	—>	**steigender Schwierigkeitsgrad**
Training	Sprecher-abhängig	Sprecher-unabhängig
Vokabulargröße	10+	1000+
Äußerungsmodus	Einzelwort	Kontinuierlich
Sprachäußerung	Gelesen	Spontan
Sprachschatz	Begrenzt	Natürliche Sprache
Umgebung	Vorteilhaft	Ungünstig

Tabelle 2.2: Eigenschaften maschineller Spracherkennungssysteme (n. [29])

Natürliche Sprache in *freier* Umgebung — ohne besondere Gewährleistung idealer akustischer Voraussetzungen — stellt damit die höchsten Anforderungen an ein maschinelles Erkennungssystem. Die prinzipielle Vorgehensweise dieser lässt sich in etwa wie folgt zusammenfassen: Basis der automatischen Spracherkennung ist die Erkennung von phonetischen Grundeinheiten, den *Phonemen*. Ein Phonem lässt sich nach [82] definieren als:

Definition 5 (Phonem) *Die kleinste bedeutungsunterscheidende lautliche Basiseinheit (Phon, Laut) ist das Phonem.*

Diese Phoneme können nun je nach Erzeugungsvorschrift[2] in unterschiedliche systematische Einteilungen gruppiert werden. Sind alle Phoneme erkannt und zugeordnet, so unterscheiden die Phoneme den semantischen Inhalt bei ansonsten gleicher Phonstruktur.

Dazu wird in einem ersten Schritt das analoge Sprachsignal bei der Aufnahme soweit als möglich von Störsignalen (Rauschen u.Ä.) befreit und mit

[2]Es finden unterschiedliche Erzeugungsvorschriften Anwendung: vgl. Amerikanischer und europäischer Strukturalismus.

einem einheitlichen Pegel aufgenommen. Anschließend erfolgt eine Diskretisierung des analogen Signals. Aus den digitalen Informationen werden sogenannte *Merkmalsvektoren* gewonnen, indem beispielsweise das Gesamtsignal in verschiedenen adäquaten[3] Frequenzbändern untersucht wird. Über kurze Zeiteinheiten (im Millisekundenbereich) werden durch eingesetzte Frequenzfilter Profile des Signals über diesen Zeitintervallen gebildet. Diese Profile, ihre Änderung in der Zeit sowie weitere Parameter (etwa Sprachenergie) bilden die Merkmalsvektoren. Letztere müssen nun mit den durch Trainingssätze gewonnenen Templates über die Zuordnung phonetischer Grundeinheiten — der Phoneme — zu Mengen von Merkmalsvektoren verglichen werden.

Eine solche sehr vereinfachte Darstellung ist in der Praxis ungenügend. Durch Unterschiede etwa in der Sprechgeschwindigkeit oder durch Co-Artikulationseffekte (auftretende Ausspracheunterschiede je nach phonetischem Kontext) müssen elaboriertere Verfahren angewendet werden. Eine prominente Vorgehensweise verwendet Hidden Markov Modelle (HMM). In diesen wird jede phonetische Einheit durch diese Einheit konstituierende Merkmalsvektoren repräsentiert und jedem Merkmalsvektor wird eine Menge von Zuständen zugeordnet. Übergänge zwischen diesen Zuständen sind mit einer Wahrscheinlichkeit versehen. Dieses Vorgehen ermöglicht es, die Wahrscheinlichkeiten der Zustandsänderungen aus Trainingsdaten zu ermitteln und während des Betriebs eine probabilistische Bewertung durchzuführen. Je nach existierenden Kanten können während der Traversierung sowohl gleiche Zustände wiederholt als auch andere einfach übersprungen werden. Aus der ebenfalls mit Übergangswahrscheinlichkeiten versehenen Komposition der verschiedenen HMM-Teilmodelle ergibt sich so ein Gesamt-Übergangsnetzwerk.

Die Kombinationen phonetischer Grundeinheiten repräsentieren die einzelnen Wörter. Diese Zuordnungen Wort-zu-Phonemkombination werden in Lexika gespeichert und müssen nun während der Graphtraversierung gefunden werden. Auch hier ergeben sich verschiedene Konstellationen, da ja schon auf phonetischer Ebene unterschiedliche Kombinationen und Sequenzen ermittelt wurden. Das Resultat besteht in einem weiteren Graphen unterschiedlicher Wortkombinationen. Zusätzliches Wissen, zum Beispiel linguistische Information über grammatikalische Strukturen, dient auf dieser Ebene der Festlegung des wahrscheinlichsten Pfades durch den gebildeten Graphen, und damit dem eigentlichen Erkennungsresultat, der ermittelten symbolischen Notation der Information aus dem analogen Sprachsignal.

[3] Zum Beispiel angelehnt durch die biologisch gegebene Frequenzsensibilität des menschlichen Gehörs.

2.2.5 Verfahren maschineller Gestenerkennung

Zur Durchführung einer Analyse multimodaler Äußerungen, müssen auf unterer Ebene Verfahren eingesetzt werden, die beiden Modalitäten Sprache und Gestik signaltechnisch zu erfassen. Auf dieser Basis können dann Aussagen über die vorliegende Äußerung getroffen werden. Hier handelt es sich, wie in Abschnitt 2.2.4 erläutert, um ein Klassifikationsproblem. In der Spracherkennung ist es das Ziel, anhand der eingegangenen Informationen über das physikalische Sprachsignal die korrespondierenden Symbole (Worte) aus der natürlichen Sprache zu finden, welche mit einzelnen Signalteilstücken korrespondieren. In der Gestenerkennung findet sich auf dieser Ebene ein erster wichtiger Unterschied, nämlich die Annahme der Existenz einer sprachlichen Grundlage, also dem Vorhandensein durch Regeln (Grammatiken) geordneter diskreter Symbole. Dennoch ist die Möglichkeit einer symbolischen Notation gestischer Äußerung in verschiedenen Gestennotationssystemen exploriert worden. Auf dieser Annahme aufbauend findet sich in [26] der Ansatz, HamNoSys[4] als Basis für eine Gestenerkennung zu verwenden. In der Art eines kompositionellen Ansatzes definieren hier Sequenzen einzelner Elemente — der HamNoSys Zeichen — als gestische Grundbausteine kombinatorisch gestische Äußerungen.

Die in der Spracherkennung eingesetzten Verfahren, um eine Abbildung zwischen Signal und Symbol zu finden, bedienen sich größtenteils statistischer Ansätze. So zum Beispiel über den Einsatz von HMMs, um das gestellte Klassifikationproblem zu lösen. Daher wurden diese Verfahren ebenfalls auf den Bereich der Gestenerkennung übertragen. Hier wurden HMM-basierte Ansätze schon teilweise sehr erfolgreich für eine Erkennung solcher gestischen Symbole eingesetzt, wie sie beispielsweise in den verschiedenen Gebärdensprachen existieren. Eine alternative Technik zu einer Modellierung über HMMs bietet der Einsatz von neuronalen Netzen. Auch hier wird eine Klassifikation über eine Menge von Eingabevektoren ausgeführt; der Unterschied besteht darin, dass in letzteren die Übergangswahrscheinlichkeiten implizit in der Parametrisierung der einzelnen Netzgewichte konstituiert sind. In beiden Ansätzen dient eine Trainingsphase entweder der Feststellung der Übergangswahrscheinlichkeiten in den HMMs oder der Adaption der entsprechender Gewichte in den neuronalen Netzen. Eine gute Übersicht über aktuelle Arbeiten auf diesem Gebiet ist in [11] und [95] zu finden. Besonders häufig werden diese an den Erkenntnissen und Methoden der Spracherkennung orientierten Vorgehensweisen bei der Erkennung von (Teilen von) Zeichensprachen eingesetzt, also einem Einsatzgebiet mit offensichtlich ähnlichen Anforderungen.

[4] Hamburger Notationssystem

Dazu alternative Vorgehen beruhen nicht auf vorherigen Trainingsdurchläufen, sondern beschreiben Gesten durch eindeutig definierende Merkmale [87] [100]. Das Wissen über den Zusammenhang zwischen Eingaberohdaten, bestimmten Merkmalen und der vorliegenden Geste wird hier nicht implizit über einen Trainingsvorgang erworben, sondern explizit im Vorfeld in kombinatorischen Regeln beschrieben. Bei gegebenen Merkmalen wird daraufhin in einem Matchingprozess auf die Erfüllung dieser Regeln geprüft.

Unabhängig davon gibt es sehr vielversprechende Ansätze, welche Gesten nicht als Ganzes klassifizieren wollen. Hier wird mittels neuronaler Methoden primär erst einmal die Position bedeutungsvoller Körperstellen, also die die Gestik begründende Körperkonfiguration bestimmt [75][76][77] und [78]. Diese Verfahren streben insbesondere den Verzicht auf herkömmliche an den Körper gebundene Sensorikmethoden an. Grundlage dafür ist die Annahme, dass die Körperkonfiguration durch die Ermittlung bestimmter Ankerpunkte des Körpers hergeleitet werden kann oder dass besonders die zu bestimmenden Ankerpunkte wichtig im Sinne einer Interpretation gestischer Äußerung sind. Diese Vorhaben streben an, die für etwaige höhere Interpretationsebenen benötigten Rohdaten bereitzustellen.

Eine früher Ansatz zur Auswertung nicht nur von Gesten mit starkem symbolischen Charakter, sondern zur Erkennung definierter geometrischer Grundmuster in den Bewegung der Extremitäten, finden wir in den Arbeiten von Koons, Sparrel und Thorisson [45][90] bei der Einführung des *Gestlet*-Begriffs. In diesen Arbeiten wird auf bereits durch Sensorik bereitgestellten Rohdaten aufgesetzt, indem auf ihnen einfache Gruppierungen nach bestimmten analytischen Ähnlichkeitsbeziehungen vorgenommen werden, um daraufhin wesentliche geometrische Formen zu ermitteln:

> *„Gestlets ... are a second layer of abstraction above raw gesture data that group together similar portions of the raw data stream.“*

Die Gestlets sind dabei reine Bewegungssegmente mit ähnlichem Bewegungsmuster. Die Grenzen zwischen diesen Gestlets werden durch die Ruhephasen in der Bewegung definiert. Gestlets abstrahieren von den einzelnen zugrundeliegenden Sensordaten und fassen sie gemäß gewissen Ähnlichkeitskriterien zusammen. Eine endgültige Interpretation der damit geäußerten Geste wird in einem weiteren Schritt über die Integration mit sprachlichen Anweisungen vorgenommen.

2.3 Virtuelle Realität

In diesem Abschnitt wird ein Überblick über die Methoden der Virtuellen Realität gegeben. Dieser beginnt mit einer formalen Einführung des VR-Begriffs. Daraufhin schließt sich eine Vorstellung bisher entwickelter konzeptioneller und technischer Modelle sowie von Besonderheiten der Datenorganisation und der Applikationslogik im VR-Kontext an. Die Vorstellung einer speziellen VR-Anwendung zur Virtuellen Konstruktion dient dazu, um den Rahmen der hier ebenfalls erbrachten praktischen Exploration anhand einer realen Implementation abzustecken. Der Begriff der Virtuellen Realität kann in vielerlei Hinsicht interpretiert und definiert werden. Für eine Betrachtung des Einsatzes multimodaler Interaktion in VR-Systemen sollen die grundlegenden Kriterien dieser Systeme insbesondere unter zwei Aspekten ausgeleuchtet werden:

- Adäquatheit multimodaler Interaktion in der Anwendungsdomäne der Virtuellen Realität.
- Besondere technische Voraussetzungen, Modellierungsmethoden und Randbedingungen in der VR-Domäne.

Eine erste Erläuterung der Adäquatheit von multimodaler MMK in der VR wurde bereits in der Motivation (s. Abschnitt 1.1.3) begonnen. Hier soll dieses noch einmal aufgegriffen werden sowie mit Hilfe von in der Literatur zu findenden Definitionen über das *Wesen* der VR eine Basis für die noch folgende Diskussion der Nützlichkeit multimodaler Schnittstellen in der VR explizit geschaffen werden. Der VR-Begriff selbst wird unterschiedlich ausgelegt. Eine umgangssprachlich Begriffsumschreibung stammt von Sherman:

„*Virtual Reality allows you to explore a computer generated world by actually being in it*“ [84].

Eine etwas formalere Definition ist die von Manetta und Blade aus einem Glossery zur VR-Terminologie:

Definition 6 (Virtual Reality:) *A computer system used to create an artificial world in which the user has the impression of being in that world and with the ability to navigate through the world and manipulate objects in the world* [62].

Beiden Definitionen gemeinsam ist der Bezug auf die technische Realisierung mit Hilfe von Computersystemen sowie der Aspekt des "*being in*", also der subjektiven Versetzung eines Benutzers in eine künstliche, nicht real existierende Welt. Inwieweit dieses gelingt, also der Benutzer die virtuelle Realität als Realität akzeptiert, wird mit dem bereits mehrfach betonten Immersionsfaktor beschrieben. Definition 6 betont zusätzlich die Möglichkeit der Interaktion mit der fiktiven Welt oder Umgebung im Hinblick auf den eigenen Standpunkt, also die Navigation, sowie die Möglichkeit der Manipulation dort erscheinender Objekte. Eine formalere Definition des Begriffes der Virtuellen Realität findet sich im M_RIC[5] [92]. Sie betrachtet den Begriff unter dem Aspekt der *kognitiven Abweichung*. Darin wird Realität als eine Menge von Interaktionen zwischen einer menschlichen Person und ihrer Umwelt gesehen. Eine Menge kognitiver Abweichungen ermöglicht nun einen Übergang von Realität hin zu virtueller Realität. Eine darauf folgende Unterscheidung von *Quellen der Virtualität* in zwei Typen betrachtet die Faktoren der Simulation in Bezug auf ihr reales Verhalten: Wird eine Übertragung der realen (physikalischen) Verhältnisse angestrebt oder werden gerade Modifikationen dieser zugelassen?

Definition 7 (Virtual Reality) *Let us take a situation V to be examined. This situation must conform with the M_RIC model, and will be called a situation of Virtual Reality if and only if its agents differ from corresponding real one(s) and if all such agents (then called "vitual agents") can be obtained from real ones thanks to virtuality sources of type 1 and/or 2* [92].

Mit dem Agentenbegriff wird hier eine Unterteilung in verschiedene Stufen im Informationsfluss zwischen dem Verstand des Benutzers und seiner Umwelt vollzogen. Dieses Modell betrachtet insbesondere Virtuelle Realität als eine spezielle Art Kommunikation mit einer artifiziellen (Teil-) Umgebung. Allen Definitionen gemeinsam ist also die Einbindung des Benutzers in einen solchen Kommunikationsfluss. Wird dieses in Definition 7 noch explizit gemacht, so findet es sich implizit in der Beschreibung von Sherman und der Definition 6 in den Begriffen "*explore, navigate*" und "*manipulate*".

Nach diesen sehr grundlegenden Betrachtungen folgen nun Ausführungen zu den technischen Grundlagen von VR-Systemen und den hier zu unterscheidenden Kategorien. Virtuelle Realität umfasst allgemein Verfahren mit deren Hilfe die Ergebnisse von Simulationen einem Anwender unmittelbar und interaktiv über dessen Sensorik zugänglich gemacht werden. Dieses betrifft in erster Linie die echtzeitfähige und möglichst realistisch wirkende graphische

[5] Französisches Akronym für "*Model Representing Cognitive x Interaction*"

Präsentation der virtuellen Umgebungen und die Bereitstellung von Interaktionsmöglichkeiten in diesen Umgebungen. Um dieses umzusetzen haben sich in den letzten Jahren generelle Standards (vgl. z.Bsp. [15][18][73] und [79]) für die strukturierte Umsetzung der dabei anfallenden Aufgabenteilgebiete herauskristallisiert, deren Grundlagen im Weiteren ausgeführt werden sollen. Diese Informationen werden sich in der folgenden Konzeption der Integration einer multimodalen, aber vor allem einer gestischen Eingabe besonders zur Erläuterung des noch auszuführenden Erweiterungsvorschlags der Szenen- und Applikationsgraphstruktur als nötig erweisen.

Geometriespezifikation und Szenengraph

Die in den heutigen Systemen für eine computergraphische Darstellung am häufigsten zu findende Modellierung der Szene und der darin befindlichen Objekte geschieht durch die Verwendung von grenzflächenorientierten und durch Polygone beschriebene Modelle und deren Anordnung in einem *Szenengraphen*, wie von Strauss und Carey 1992 [89] vorgeschlagen. Diese Form der Strukturierung arbeitet objektzentriert und erlaubt das nachträgliche Bearbeiten und Verändern der repräsentierten Objektattribute.

Grundsätzlich wird in diesen Systemen auf der untersten Ebene, der Ebene der Geometriebeschreibung, ein Polygon durch die Verbindung und Kantenbildung zwischen Knoten — den 3D-Vektoren — gebildet. Die Polygone wiederum definieren, zusammen mit zusätzlich angegebenen Normalenvektoren, die Flächen, welche damit nun die benötigten Informationen zu einer Beleuchtungsberechnung, dem *Shading* besitzen. Durch die Verwendung von Normalen können insbesondere sehr effiziente und dabei in Grenzen realistisch wirkende Renderingverfahren, z.Bsp. *Gouraud-Shading* [25], eingesetzt werden. Ein gängiges Hilfsmittel zur Erhöhung des Realismus in der Darstellung besteht in der Verwendung von *Texturen*, also von Bildinformationen, welche auf die Flächen projiziert werden und welche der in der Realität erwarteten Oberflächengestaltung gleichen. Um modulare und effiziente Methoden zur Kapselung und Manipulation dieser Geometriebeschreibung zu gewährleisten, werden diese in einem speziellen Typus eines *nodes* (Knoten) aggregiert. Diese Objekte werden anschließend in einen Szenengraphen eingehängt und verwaltet. In dieser Graphstruktur sind Knoten verschiedenen Typs so untereinander verbunden, dass durch die Kantentraversion zwischen den Nodes spezifische Beziehungen ausgedrückt werden.

Beispiele für die Bereitstellung von Verfahren, um solche Graphstrukturen zu modellieren, sind der für die Verwendung im WWW (World Wide Web) konzipierte Standard VRML (Virtual Reality Modelling Language) —

jetzt in der Version VRML97 [15] — oder die von der Firma SGI kommerziell vertriebenen Graphikbibliotheken Open Inventor [99] und Performer [83] sowie die jüngsten Bestrebungen eines offenen Standards einer Szenengraphbeschreibung durch das *OSG*-Konsortium (OSG: Open Scene Graph) [80]. Daneben existieren diverse Werkzeuge zur graphischen Erstellung und Erzeugung[6] der zugrunde liegenden Polygonmodelle.

Positionierung und Ausrichtung der einzelnen Polygone geschieht mittels Methoden der Linearen Algebra, durch die Multiplikation mit homogenen 4x4 Matrizen, den CGMs (Computer Graphics Matrices). Die wichtige Eigenschaft der Konkatenierungsmöglichkeit der CGMs erlaubt eine hierarchische Anordnung von Objekten und durch einfache Matrixmultiplikation den Aufbau von relativen Lage- und Ausrichtungsbeschreibungen innerhalb des Szenengraphen. Dieser bildet so einen Baum, in welchem die einzelnen Basisknoten als Container unterschiedliche Aufgaben haben können. Allen diesen auf einer Szenengraphmodellierung operierenden Systemen gemeinsam sind die hier zu betrachtenden Knotentypen *CGM*, *Gruppe* und *Geometrie*. Die Geometrieknoten definieren die Vektoren der Polygone in ihrem lokalen Ursprungssystem und stellen im Baum die Blätter dar. Gruppenknoten vereinen Teilbäume unter einem Knoten im Sinne einer Aggregation. Die CGM-Knoten definieren jeweils ein eigenes lokales Koordinatensystem, in welchem alle darunter befindlichen Teilbäume definiert sind. Diese CGMs werden während der sogenannten Traversierung des Baums aufmultipliziert und ermöglichen so die beschriebenen relativen Positions- und Lageangaben.

Diese Art der Szenenspezifikation in einer Graphstruktur ist durch diesen impliziten "Vererbungsmechanismus" — auch Materialeigenschaften können so vererbt werden — als die vorrangige Struktur zur Definition computergraphischer Szenen anzusehen. Viele *Renderengines*, die nicht objektzentriert arbeitenden Schnittstellen zur Graphikhardware, können die Ergebnisse während des Traversierens direkt in ihre eigenen Datenstrukturen übernehmen (s. State-Machine in [5]). Neben den hier besprochenen Knotentypen gibt es je nach System viele, auch grundsätzlich andere Arten von Nodes. In VRML97 haben so beispielsweise auch Knoten zur Modellierung von bewegten Eigenschaften und Ereignissen Einzug gehalten, um die Szenen dynamisch und interaktiv zu gestalten. Dieses muss nicht zwangsläufig als Definition im Szenengraph erfolgen, im Sinne eines Standards und einer Portabilität dieser Eigenschaften ist es aber ein Bestreben, diesen eigentlich anwendungs- oder simulationsbezogenen Teil in eine gemeinsame Strukturbeschreibung zu übernehmen. Es obliegt dann beispielsweise dem VRML-Interpreterprogramm

[6] Vgl. dazu z.Bsp. die kommerziellen Systeme: Softimage, LightWave, 3D Studio etc.

— dem Viewer — diese enthaltenen Informationen auszulesen und umzusetzen.

Szenensicht und deren Projektion

Wie bereits in Abschnitt 1.1.2 erläutert wurde, beruht die Immersion primär auf kontinuierlicher stereoskopischer Berechnung der visuellen Szenensicht relativ zu einem Betrachterstandpunkt. Die jeweilige Projektion, der auf eine Oberfläche abgebildete Blick auf die Szene, wird dabei durch das *Viewing-Frustum* definiert (s.a. [25]). Dieses ist eine elaboriertere "Sicht"-Definition als sie beispielsweise durch eine Kamera-Metapher gegeben ist, erstere stellt alle benötigten Parameter für verschiedene Projektionsmodelle zur Verfügung. Zwei verschiedene Arten der immersiven Darstellung können grundsätzlich unterschieden werden. Bei den Head Mounted Displays (HMDs) wird das Viewing-Frustum als virtuelle Kamera mit der Blickrichtung des Betrachters bewegt. Auf starr vor den Augen befestigten Displays wird eine immer gleichwinklige — gemessen zwischen Blickrichtung und Projektionsfläche — Projektion der Szene erzeugt. Dagegen sind bei den Großdisplays, etwa einer Wall oder Cave, die Kopfrichtung und Ausrichtung der Projektionsflächen entkoppelt; für die Bildberechnung wichtig ist hier die relative Lage und Ausrichtung des Kopfes in Bezug zu den Projektionsflächen und die relative Lage dieser in der virtuellen Szene. Gemeinsam ist beiden Modellen natürlich die Bedingung zur kontinuierlichen Neuberechnung der Szene und der jeweiligen Sichtparameter, des aktuellen Viewing-Frustums.

Eingabegeräte und Sensorik

Um mit der dargebotenen Szene zu interagieren, wurden diverse Eingabegeräte entwickelt [58][85]. Ein guter Überblick über Sensoriken mit spezieller Eignung für die Erfassung von menschlichen Bewegungen findet sich auch in [72]. Einfache desktopbasierte VR-Systeme bedienen sich fast ausschließlich der Tastatur und der Maus. Letztere wurde zur *Space-Mouse* weiterentwickelt, um die in 3D vorhandenen zusätzlichen DOFs (Degrees Of Freedom) zu manipulieren.

Der *Stylus* ist dagegen eine Art stiftbasiertes Eingabegerät. Es ist wie die 2D/3D Mäuse ebenfalls mit einem oder mehreren Schaltern zur Vermittlung diskreter Eingabewünsche (dem Mausclick) ausgestattet. Während der Benutzung wird an seiner Vorderseite einen Zeigestrahl in die virtuelle Szene projiziert. Mit Hilfe dieses Strahls kann ein gewünschtes Objekt in der Szene angezeigt werden. Man bringt Objekt und Strahl zur Deckung und betätigt daraufhin einen der Schalter.

Mit einem *Data-Glove* können dagegen ganze Handposturen sehr genau und mit hoher Auflösung registriert werden. Überträgt man diese Informationen in die Szene, zum Beispiel in ein virtuelles Counterpart der Hand, so können diese Geräte zum Fassen von Objekten benutzt werden. Man überprüft, ob das graphische Handmodell Kontakt zu einem Szenenobjekt hat und ob die Hand *geschlossen* ist. Trifft beides zu, so kann ein Zufassen — die Auswahl des Objekts — gefolgert werden.

Bei den letzten beiden Eingabegeräten wurde der Bedarf nach Positions- und Lagebestimmung noch nicht angesprochen. Zur sinnvollen Benutzung des Stylus als auch der Gloves ist dieses eine zwingende Voraussetzung. Hier benötigen wir Möglichkeiten die 6DOFs, welcher ein freier fester Körper in 3D besitzt, zu erfassen, das so genannte *Tracking*. Unterscheiden kann man prinzipiell vier verschiedene Arten von Sensoren, welche diese Informationen liefern:

- Optisches Tracking
- Akustisches Tracking
- Elektromagnetisches Tracking
- Gyroskopische Verfahren

Bei allen vier Verfahren ist es die Aufgabe, vorher definierte Ankerpunkte auf Eingabedevices und Körpern während einer kontinuierlichen Bewegung zu verfolgen und sowohl die neue Position als auch die neue Lage in Bezug zu einem festen Koordinatensystem zu ermitteln. Optisches Tracking verwendet Kameras (mitunter mehrere verschiedene, um eine 3D-Berechnung zu erleichtern), um entweder die Ziele direkt oder auf ihnen befindliche Marker zu finden. Akustisches Tracking bedient sich des Ultraschalls. Elektromagnetische Verfahren erkennen die Bewegungen anhand von Induktion in Sensoren während der Bewegung durch ein Magnetfeld. Gyroskopische Verfahren kontrollieren die relativen Beschleunigungen nach einer initialen Position und schließen so auf die aktuellen Informationen. Eine prinzipielle Eigenschaft dieser Sensoren und der Gloves ist die Art ihrer Datenweitergabe. Diese erfolgt in einem eigenen Takt (Samplingrate) der entsprechenden hardwaretechnischen Realisierung. Diese Eingabegeräte erlauben die kontinuierliche Auswertung der Position und der Lage sowohl der Eingabegeräte als auch der Benutzersicht, indem der Kopf des Benutzers getrackt wird.

Ereignisverarbeitung und Render-Loop

Der technische Vorgang der Neuberechnung der jeweiligen Projektion wird in der Render-Loop [25] vollzogen. In dieser werden prinzipiell die einzelnen Phasen *App* (Application), *Cull* (Culling) und *Draw* unterschieden. Diese prinzipielle Unterteilung in Phasen basiert auf der Umsetzung des Prinzips der Rendering Pipeline [25] und berücksichtigt allein die Aufbereitung der graphischen Szene mit den benötigten Schritten vom 3D-Modell bis zur Wiedergabe auf einem Projektionsgerät.

In diesen Phasen werden verschiedene Teilaufgaben bearbeitet. Die App-Phase verankert den applikationsspezifischen Teil, die eigentliche Anwendungssimulation mit der Generierung der neuen Szene. Hier geschieht die Auswertung der Eingabedevicedaten, die Festlegung der neuen Benutzersicht oder die Auslösung von anderen, den Eingaben entsprechenden Aktionen. Die Cull-Phase sortiert vor allem die außerhalb des nun aktuellen Viewing-Frustums liegende, also die nicht sichtbare Geometrie aus. Die Draw-Phase letztendlich sorgt für die eigentliche Transformation vom 3D-Koordinatenraum auf den 2D-Displayraum des Darstellungsgerätes und die Applikation der Beleuchtungsbeschreibung — das Rendering.

Die Verarbeitung der Eingabedaten erfolgt in der Art der Ereignisauswertung, wie sie auch bei den 2D-graphischen Fensteroberflächen Verwendung findet. Eine Message-Loop arbeitet alle eingehenden Daten kontinuierlich aus, entsprechende Aktion erfolgen nach dieser Auswertung. Die App-Phase der Render-Loop ist das entsprechende Pendant in einem VR-System. Die Schwierigkeit bei diesem Vorgehen ist die Bedingung, in dieser App-Phase nicht mehr Zeit zu verbringen als eine kontinuierliche (mindestens 20-25 Frames/sec) Anzeige notwendig macht. Diese Gewährleistung der Framerate wird erschwert durch die nur schwer abzuschätzende Laufzeit der Cull- und Draw-Phasen. Anwendungspezifische Aktionen und Interaktionen einer Implementation finden grundsätzlich in der App-Phase statt. Wie dieses geschieht, wird nicht weiter spezifiziert, es gibt keine Methodenvereinbarung für den benötigten Aktionsfluss.

Spezifikation von Applikationslogik

Um das Manko der nicht vorhandenen Möglichkeit zur offenen Spezifikation von Applikationslogik in vielen VR-Systemen zu beseitigen, wurden verschiedene Bestrebungen unternommen. Eine der bekanntesten ist die mit der VRML97 Spezifikation verabschiedete standardisierte Definition verschiedener Knotenklassen zur Berücksichtigung von aktiven und interaktiven Si-

mulationseigenschaften durch die Einführung von *Engines* und *Sensoren*. Viele der dort entwickelten Ideen wurden ebenfalls in die Java 3D und X3D [97] Programmierschnittstellen übernommen. Diese APIs sind Beispiele der Bereitstellung von Erweiterungen über eine Angabe von reiner Geometriebeschreibung hinaus. Sie gestatten weiterhin eine weitreichende Spezifikation der eigentlichen Viewdefinition, von dynamischen Vorgängen und Abläufen in der Szene, sowie der Möglichkeit einer gewissen Abstraktionen von den eingesetzten Eingabegeräten, und betten diese Möglichkeiten ebenfalls in den Szenengraphen ein. Mit den zuletzt genannten Möglichkeiten wird damit auch ein Teil der eigentlichen Applikationslogik in den strukturellen Aufbau des Graphen verlagert und damit portierbar gestaltet, eine Grundvoraussetzung für den Einsatz im WWW. Ähnliche Ansätze findet man auch bei Vertretern erweiterter VR-Autorensystems, zum Beispiel dem am Fraunhofer Institut entstandenen *Lightning*, dem *DIVE*[7] System, bei *Massive*-(1, 2 oder 3)[8] oder den kommerziellen Systemen wie dem *World Toolkit* von Sense 8 oder *Superscape* von Superscape Ltd.

Das von der GMD[9] entwickelte *AVANGO*[10] [91] findet hier Interesse, da es sich besonders eignet, um den implementatorischen Teil der hier noch auszuarbeitenden Konzepte für eine multimodale Interaktion umzusetzen. AVANGO erweitert eine der ursprünglichen VR-Graphikbibliotheken, das bereits erwähnte Performer-Toolkit, und kann sich so dessen Möglichkeiten der Echtzeitfähigkeit und die dort zur Verfügung stehenden grundlegenden Szenengraphprimitiven zunutze machen. Dabei werden die zur Verfügung gestellten Klassen um zwei funktionale Merkmale angereichert: Die Objekte erhalten eine Schnittstelle zu einer *Scheme* Programmierumgebung und ermöglichen so auch schnelle prototypische Implementationen. Noch interessanter ist die Einführung von *Fieldcontainern* (vgl. VRML97) — Konzepte, welche den Datenaustausch zwischen Objektknoten auch orthogonal zur Szenengraphstruktur und dessen Traversierung gewährleisten. Miteinander verbundene Fields (Felder) übermitteln Daten zwischen den Knoten, in AVANGO auch prozess- und netzwerkübergreifend, und ermöglichen so die Spezifikation von Applikationslogik innerhalb der Graphstruktur. Durch die Opazität dieses Datenaustauschs zwischen den Fields ist es darüber hinaus möglich, verteilte VR-Anwendungen zu entwickeln: Durch welches Medi-

[7] Distributed Interactive Virtual Environment - Entwickelt am Swedish Institute of Computer Science.

[8] Entwickelt von der Communications Research Group, School of Computer Science and Information Technology, University of Nottingham.

[9] Gesellschaft für Mathematik und Datenverarbeitung.

[10] Ehemals *AVOCADO*.

um — ob lokal über einen gemeinsamen Speicherbereich oder global basierend auf Netzwerkprotokollen — dieser Datenaustausch geschieht, ist für eine funktionale Beschreibung des logischen Zusammenhangs nicht wichtig. Die Auswertung der *Fieldconnections* wird automatisch während der App-Phase durchgeführt, ihre Spezifikation kann aber vorher statisch geschehen. Diese Art der Informationverarbeitung ist bei der noch folgenden Konzeption, vor allem der Umsetzung der gestischen Interaktion, besonders zu betonen.

2.3.1 Virtuelle Konstruktion

Virtuelle Konstruktion (VK) stellt ein Spezialgebiet einer Anwendung der Virtuellen Realität dar. Hier werden rein geometriebasierte Operationen von CAD-Systemen um Konstruktionswissen über die bearbeiteten Objekte, Baugruppen und Verbindungsmöglichkeiten angereichert, um zusätzliche funktionale Eigenschaften zu gewährleisten. Ziel ist eine Verbesserung der Planung und Interaktion während der Konstruktionsphase; weg vom computergraphisch-zentrierten Denken über die geschickte Kombination von graphischen Primitiven (Punkte, Linien, Dreiecke, Splines, Quader ...) zu immer komplexeren Objekten und hin zu einer wissensgestützen Montage mit Entitäten aus der entsprechenden Konstruktionsdomäne (z.Bsp. Flugzeug-/Fahrzeugbau), welche als benennbare Entitäten mit eindeutigen über die Geometrie hinausgehenden Eigenschaften repräsentiert sind: Erfüllen Objekte in zusammengesetzten Aggregaten bestimmte Funktionen oder haben sie ersichtliche Eigenschaften wie Farbe oder Form, so können sie beispielsweise sprachlich benannt oder beschrieben und so anschließend für einen Arbeitsschritt ausgewählt werden. Anstatt im CAD-System geometrische Primitive exakt räumlich anzuordnen und anschließend zu verknüpfen, können modellierte Verbindungsstellen auch bei vager Annäherung "zusammenschnappen" und anschließend den Verbindungseigenschaften entsprechenden Bewegungseinschränkungen unterliegen und vieles mehr.

Die Arbeitsweise, die Begrifflichkeit und die Terminologie entsprechen dagegen bei einem CAD-Einsatz der der gewählten Repräsentationstechnik. Ein Benutzer muss neben seiner Fachkompetenz und Expertise über das Konstruktionsziel ebenso die speziellen Handlungsabläufe in der CAD-Anwendung kennen und seine ingenieurseigene "Konstruktionswelt" in die computerinterne "Repräsentationswelt" übertragen. Hier bietet die VK eine geeignete Schnittstelle, um die gewünschte Umsetzung aus Konstruktionssicht zu realisieren: Die einem Anwender zugänglichen Objekte sind z.Bsp. nicht geometrische Grundprimitive sondern im Erscheinungsbild und Größe parametrisierte Werkstoffprimitive wie Schrauben, Leisten, Rohre, Winkel etc.

Die folgenden Abschnitte geben eine Einblick in das in dieser Arbeit betrachtete System zur VK und stellen die Arten der von einer Interfacekomponente zu realisierenden Interaktionen vor. Dabei wird besonderer Wert auf Interaktionsmetaphern und -möglichkeiten gelegt, welche nicht nur spezifisch für die VK sind, sondern welche allgemein für VR-Anwendung geeignet erscheinen.

Der CODY - Virtuelle Konstrukteur

Diese Anwendung [14][39][41] bietet über die wissensbasierte Anreicherung der verwendeten Objekte und Konzepte Möglichkeiten, den Konstruktionsprozess intelligent zu unterstützen. Über ein Mausinterface und eine natürlichsprachliche Tastatureingabe werden im bisherigen System zwei verschiedene Interaktionsformen zur Verfügung gestellt [34]. Die Texteingabe ermöglicht die direkte Eingabe eines Interaktionswunsches über die Tastatur. Der Benutzer kann mit ihrer Hilfe Aktionen und Objekte sprachlich spezifizieren (Bsp.: *"Verbinde das obere rechte Rad mit der gelben Leiste"*) und in einem Schritt umsetzen lassen. Dagegen manipuliert die Mauseingabe interaktiv die dargestellte Szene. Aktionen werden über eine Menüleiste ausgewählt und die Zielobjekte mit dem Mauscursor ausgewählt. Gemäß dieser Aktion wird die Manipulation nun durch Mausbewegungen durchgeführt, zwei Objekte werden beispielsweise nach der Spezifikation des Verbindungsmodus direkt zum Kontakt und damit zum "dauerhaften" Schluss gebracht. Der Konstruktionsprozess kann dabei während der Interaktion in einer graphischen Perspektivprojektion verfolgt werden. Die folgenden Abschnitte erläutern die für die Systemfunktionalität besonders wichtigen Wissensstrukturen.

Konstruktionswissen: Der COAR-Formalismus (COAR: Concepts for Objects, Assemblies and Roles) [38] ist ist eine framebasierte Wissensrepräsentation, in welcher benötigtes Konstruktionswissen im VK implementiert wird und einzelne Grundbauteile mit ihren Attributen spezifiziert werden. Werden Objekte nun zusammengesetzt, erfüllen sie in den resultierenden Aggregaten für diese spezifische Rollen, indem sie beispielsweise geometrische Constraints (Bedingungen) erfüllen. Sind die Bedingungen im Laufe eines Konstruktionsvorgangs gegeben, wird ein neues Aggregat gemäß seines Konzepts in der Wissensbasis instanziiert und dient nun als weitere Baueinheit. Sukzessive lassen sich so aus kleinsten Einheiten komplexe Anordnungen schaffen und die Richtung des Bauvorhabens bestimmen. Die geschilderte Aggregatbildung bietet im laufenden Konstruktionsprozess die Möglichkeit, entstandene Aggregate als Ganzes verbal zu referenzieren — zu benennen, ei-

ne wichtige Voraussetzung für eine natürlich-sprachliche Eingabe. Laufende Arbeiten am CVK klassifizieren die dargestellten Objekte ebenfalls nach ihrer generischen Form und ziehen Inferenzen über Konstellationen gegebener Formprimitive zueinander [33], eine weitere Abstraktion des Konstruktionsprozesses.

Verbindungsinformationen: Um Objekte zu Aggregaten zu verbinden, müssen Informationen über mögliche Verbindungsstellen zur Verfügung gestellt werden. In [46] wird eine solche Verbindungstaxonomie entwickelt. Die sogenannten *Ports* reichern die Basisobjekte um Art und geometrische Lage der auf ihnen befindlichen Verknüpfungsstellen an. Mit Hilfe dieses Wissens können während der Interaktion Inferenzen über mögliche Verbindungsarten — neben ihrer verbalen Referenzierung — durchgeführt werden. Die Ports stellen nach einem erfolgten Portschluss, einer etablierten Verbindung zwischen zwei Ports, die nun noch zugänglichen Freiheitsgrade zur Verfügung; für das Gelingen einer Interaktionsumsetzung ist dieses eine unbedingt benötigte Information, da zu diesem Zweck beispielsweise eine noch freie Drehachse zu bestimmen, oder die Translation zu begrenzen ist. Zur Modellierung der Information über mögliche Freiheitgrade, wurde dazu in [46] das Konzept der *erweiterten Freiheits-Matrizen* entwickelt (s.a. [40]):

„Die erweiterten Freiheits-Matrizen dienen als Beschreibungsmittel, um die charakteristischen Freiheitsgrade einer Bewegung anzugeben. Sie werden verwendet, um verbleibende Relativ-Bewegungen zweier verbundener Ports oder charakteristische Freiheitsgrade verschiedener Verbindungstypen spezifizieren zu können.“ (n. [46])

2.3.2 Interaktionstypen und Manipulationsanforderungen

In den vorangegangenen Abschnitten wurden bereits exemplarisch einige speziell im text- und windowsbasierten CVK Interface implementierten Interaktionmöglichkeiten beschrieben. Viele der benötigten Operationen sind dabei in keiner Weise auf die VK beschränkt, sondern sie können als Basisoperationen für solche VR-Systeme identifiziert werden, in welchen es gilt einzelne Entitäten zu manipulieren. In einem Konstruktionsprozess bestehen zusätzlich dazu durch die Simulation gegebene Beschränkungen für die einzelnen Operationen (s. Abschnitt 2.3.1). Wie diese Beschränkungen im System während der Interaktion verankert werden, wird im Abschnitt 4.4 näher erläutert. Allgemein benötigen wir zwei grundlegende, die räumliche

Anordnung betreffende Operationen, welche sich prinzipiell nur in der Art des Zielobjekts unterscheiden, aber aus Interaktionssicht zwei grundsätzlich unterschiedliche Metaphern bedeuten:

1. Navigation: Veränderung des Betrachterstandpunkts
2. Manipulation: Veränderung der Lage und Ausrichtung von Objekten

Für den Einsatzzweck in VK muss eine besondere Möglichkeit zur Navigation nicht zwangsläufig zur Verfügung gestellt werden, da sich die zu manipulierende Szene häufig in einem räumlich begrenzten Abschnitt befindet (dem Konstruktionsraum). Auf der anderen Seite soll die dargestellte Szene aus allen Blickrichtungen inspiziert werden können. Will der Konstrukteur die Rückseite der Szene betrachten, muss er alle Objekte um den Szenenmittelpunkt drehen oder den gegebenen Dualismus ausnutzen und seinen eigenen Standpunkt hinter die Szene manövrieren. Soll dieses — unter vielem anderen — unterschieden werden, müssen grundlegend folgende Referenzen identifiziert werden:

1. Aktionsreferenzen: Identifikation auszuführender Operationen
2. Objektreferenzen: Identifikation der Zielentitäten
3. Ortsreferenzen: Identifikation zu spezifizierender Orte

Bei den Aktionsmöglichkeiten lassen sich wiederum einige VR-typische Interaktionen formulieren. Grundlegende objektzentrierte Manipulationen betreffen in erster Linie die durch die dreidimensionale Darstellung gegebenen Eigenschaften der Objekte:

1. Translation: Verschiebung der Lage
2. Rotation: Veränderung der Ausrichtung
3. Skalierung: Größenänderung
4. Deformation: Formänderung

Neben diesen im Virtuellen räumlich manifestierten Attributen stehen, je nach Qualität der Simulation, auch weitere zu verändernde Erscheinungseigenschaften als Manipulationskandidaten zur Verfügung:

1. Färbung der Objekte

2. Oberflächenmuster, Texturierung
3. Oberflächenbeschaffenheit und Struktur
4. Geräuschemission, Klangverhalten
5. ...

Diese Aufzählung stellt keinen Anspruch an Vollständigkeit; je mehr Attribute in der Simulation implementiert werden, umso mehr Möglichkeiten sie zu manipulieren scheinen wünschenswert. Besonders die räumlich verankerten Attribute sollen hier hervorgehoben werden, stellen sie durch ihre prinzipielle Nähe zu den in Abschnitt 2.2 betrachteten gestischen Benutzeräußerungen gute Voraussetzungen, um adäquate Transformationen zwischen beiden in der Interaktion zu gestatten — Diese Attribute und die Benutzergestik manifestieren sich beide im Raum. Nicht zuletzt sind sie in jeder VR-Umgebung implizit (vgl. Abschnitt 1.1.2) und daher auch für eine allgemeine Betrachtung VR-gerichteter gestischer Interaktion von Interesse.

CVK-Manipulationen: Zu den speziell im CVK zur Verfügung stehenden Interaktionen gehört die Möglichkeit zur Spezifikation einer Verbindungsetablierung oder -trennung zwischen Objekten. In Abschnitt 2.3.1 wurden drei verschiedene Verbindungsarten modelliert, besonders zu berücksichtigen sind die Einschränkungen in den möglichen Freiheitsgraden nach einer erfolgten Verbindung. Sollen nach einem Verbindungsschluss Lage-, Größen- oder Orientierungsänderungen vorgenommen werden, müssen die Einschränkungen von der Interfacekomponente berücksichtigt werden. So stellen sogenannte *point-ports*, *extrusion-ports* und *plane-ports* respektive je einen, zwei und drei Freiheitsgrade zur Verfügung. Die grundsätzliche Gewährleistung der Berücksichtigung dieser Einschränkungen hat weitreichenden Einfluss bis auf die Ebene der Gestenauswertung. Die durch die Verbindungstypen gegebenen Verbindungseigenschaften und resultierenden Freiheits-Matrizen stellen im Prinzip das Kontextwissen dar, welches während der Interpretation einer gestischen Äußerung die Erkennerresultate modifiziert. Wie die Einbindung dieses Wissens realisiert wird, ist Thema in Abschnitt 4.4.

2.4 Diskussion

In diesem Abschnitt werden die bisherigen Ergebnisse bei einem Einsatz multimodaler Eingabesysteme für die Interaktion mit Anwendungen der Computergraphik oder speziell der Virtuellen Realität untersucht und diskutiert.

So lassen sich in einer ersten Näherung die Interaktionsarten ermitteln, welche bereits durch die geschilderten Vorhaben und den damit verbundenen Implementationen validiert wurden. Der Schwerpunkt wird hier vor allen Dingen auf die Möglichkeiten einer gestischen Interaktion gelegt. Die Betrachtungen dieser Systeme gestattet weiterhin eine erste Analyse eingesetzter Techniken sowohl der Spracherkennung, der Gestenerkennung, der multimodalen Integration und Interpretation als auch in wenigen Fällen der methodischen Einbettung solcher Techniken in ein Graphik- und besonders in ein VR-System. Die Resultate der Gestenforschung und der Experimente zum Verhältnis Sprache/Gestik dienen der Vertiefung und der theoretischen Fundierung der folgenden konzeptionellen Planung einer Gestenerkennung sowie einer multimodalen Integration. Eine genauere Untersuchung zum kommunikativen Gebrauch von Gestik bewertet ebenfalls die bereits in erfolgten Umsetzungen getroffenen Entscheidungen über mögliche gestische Interaktionen. Resultierend eröffnen die Ausführungen zum Bereich der Virtuellen Realität einerseits die Nützlichkeit multimodaler Interaktion in der VR, andererseits beschreiben sie prinzipielle Schwierigkeiten einer methodischen Einbettung solcher Techniken in den Kontext des Forschungsstandes bei der Realisierung von VR-Anwendungen.

2.4.1 Möglichkeiten sprach-gestischer Interaktionen

Die grundlegende Spezifizierung benötigter Operationen in VR-basierten Systemen unterscheidet zwei Aufgaben, welche es mit einem multimodalen Interaktionssystem zu bewältigen gilt:

1. Bestimmung des oder der Referenten: Auswahl der zu manipulierenden Objekte. Die sprach-gestischen Äußerungen beziehen sich auf die *aktuelle* oder *vergangene* Konfigurationen der gezeigten Szene.

2. Bestimmung der gewünschten Veränderung. Sprach-gestische Äußerungen beziehen sich entweder

   - auf die *angestrebte* — also die veränderte — Szenenkonfiguration oder
   - auf die *Form* der Veränderung.

Eine rein gestische Erkennung kann den entsprechenden Kontext nicht eindeutig bestimmen. Formgleiche Gesten — Gesten mit gleichen spatiotemporalen Merkmalen — können unterschiedliche Bedeutung haben. Nespoulous und Lecour [74] tragen diesem Umstand Rechnung, indem sie den

Begriff Coverbale Gesten geprägt haben und die illustrativen Gesten ersteren zuordnen. Sie klassifizieren dabei den *kommunikativen Zweck* der Gesten und nicht ihre *Form* unter multimodalen Aspekten. Gerade die illustrativen Gesten dienen aber der Kommunikation spatialer Konzepte und machen sie damit für die VR-Interaktion besonders geeignet. Die in Abschnitt 2.3.2 gelisteten Interaktionstypen sowohl für die Manipulation als auch für die Navigation gründen auf der Auflösung von Aktions-, Objekt- und Ortsreferenzen. Durch die eventuelle Formgleichheit muss sich eine Gesteninterpretation auf einen anderweitig spezifizierten Kontext stützen. Dieser kann im Rahmen eines multimodalen Interaktionssystems prinzipiell durch verschiedene Quellen eröffnet werden, so durch: a) eine weitere Modalität, im hier geschilderten Vorhaben durch die Sprache, b) einen eindeutigen Anwendungszustand und c) Diskurswissen.

2.4.2 Gesten für eine multimodale Interaktion

Benutzerrelative Position

Objekt- und Ortsreferenten einer zu manipulierenden Szene sind nach Abschnitt 2.3.2 primär benötigte Informationen für die Vermittlung eines Interaktionswunsches. Gesten des Typus I, deiktische Gesten, sind dabei eine der intuitivsten Möglichkeiten der Kommunikation und finden in allen Klassifikationsversuchen einen eigenständigen Platz. Die ausdrucksstärkste Form einer deiktischen Geste ist das Zeigen auf etwas. Diese Form ist so prägnant, dass sie allein an vielen Stellen als deiktische Geste angesehen wird (auch in [74]). Das Zeigen stellt schon im Kleinkindalter eine der ersten erlernten Gestikarten dar, da dessen erster Gebrauch einhergeht mit dem Bewusstwerden des *Ich* im Gegensatz zu der Umwelt *dort* und den *darin* befindlichen Objekten relativ zum Individuum. So ist es auch erklärbar, warum Zeigen in vielen Systemen zur Referenzauflösung Verwendung findet (s. [8][17] und [57]). Daneben wird aber auch die Blickrichtung, die gesamte Körperorientierung oder das Kopfnicken in eine Richtung zur zwischenmenschlichen Kommunikation der Referenz benutzt. Von diesen findet bisher nur die Blickrichtung Interesse [9][45][57], sie wird dabei in einigen Fällen nur als nützlich bestätigt und im Hinblick auf eine Realisierung einer technischen Dereferenzierung nicht ausgewertet.

Gestisch deiktische Äußerungen sind aber unpräzise. Selbst die in ihrer räumlichen Ausformung expressivste und konkreteste Art, das Zeigen mit der Hand, führt allein durch das Setzen eines spatialen Index nicht zur eindeutigen Identifikation des/der Referenten. Es liegt hier prinzipiell eine

gewisse Unschärfe vor. In [8] wurde daher auf eine Repräsentation der Zeigerichtung mittels eines sichtbaren Cursors zurückgegriffen. Weiterhin deuten alle deiktischen Gesten nur eine Richtung vom Gestikulierenden aus gegeben an, es werden keine Tiefeninformationen übermittelt. Für einen Einsatz in VR gilt es daher zusätzlich das Problem der Tiefenstaffelung von Objekten zu berücksichtigen.

In der zwischenmenschlichen Kommunikation wird daher bei der Interpretation deiktischer Gesten sowohl Kontextwissen des laufenden Diskurs benutzt als auch sprachlich geäußerte Attributbeschreibungen der Referenzen ausgewertet, wie etwa die Farbe oder die Form von Objekten (daher *coverbal* nach [74]). Der Vorgang der Dereferenzierung greift auf Informationen aus all diesen Quellen gemeinsam zurück. Eine adäquate Modellierung einer Auswertungsfunktion mehrerer Informationsquellen, etwa Zeigen gemeinsam mit Blickrichtung und verbaler Äußerung, kann aus den untersuchten Arbeiten nicht ermittelt werden.

Aktionsbeschreibung

Kinemimische oder mimetische Gesten veranschaulichen eine Bewegung. Das dynamische Vormachen einer Rotation mit den Händen oder Armen kann beispielsweise die gewünschte Lageänderung eines Objektes veranschaulichen. Dabei lässt sich ggfs. gleichzeitig einerseits der Rotationsmittelpunkt und andererseits die Rotationsweite gestisch andeuten. Betrachten wir das Zielszenario der VK, erscheinen diese Gestentypen besonders geeignet, um gewünschte Manipulationen zumindest teilweise zu spezifizieren (vgl. dazu den Ansatz in [98], diese Gestentypen in Ansätzen zur Steuerung in einem VR-Szenario auszuwerten). Daneben ist es aber ebenso vorstellbar, dass mit der Geste die aktuelle Bewegung eines Objekts beschrieben wird, um es zu identifizieren. Auch hier liefert eine reine Gestenanalyse unzureichende Informationen für eine Interpretation (den Interaktionskontext). Eine Interaktionsmöglichkeit mit Hilfe kinemimischer oder mimetischer Gesten ist bei keiner der untersuchten Arbeiten mehr als ansatzweise verfolgt worden.

Relative Ausrichtung und Form

Die nach [74] unterschiedenen pictomimischen und spatiographischen Gesten kommunizieren spatiale Konfiguration und Form eines lexikalischen Referenten respektive. Bei dieser Art der Gestik wird die prinzipielle dimensionale Nähe zwischen dem zu kommunizierenden Faktum und der in Zeit und Raum stattfindenden Gestenexpression zunutze gemacht. Dabei kann das Faktum

auch metaphorisch spatial ausgedrückt werden (zum Beispiel wenn der Abstand der Hände eine Zeitspanne verdeutlicht, s. Tabelle 2.1 Typ V).

Diese Gestentypen werden in vielen Taxonomien als ikonische Gesten bezeichnet, letztere bezeichnet aber eher eine Superklasse pictomimischer und spatiographischer, in Ansätzen auch mimetischer Gesten (s. Tabelle 2.1 Typ II). Sparrell und Koons [87] wiederum nennen diesen Gestentypen "*depictive gestures*", sprechen im Verlauf der zitierten Arbeit aber ebenfalls vom "*iconic mapping*". Die nach [74] getroffene Unterscheidung bietet dagegen eine deutlich feinere Granularität der Taxonomie im Hinblick auf den kommunikativen Zweck der geäußerten Gestik.

Vorhaben, diese Gestentypen zur Interaktion auszuwerten, finden sich in [9], [57] und [87]. Sparrel und Koons [87] dereferenzieren Objekte und Orte (bzw. Richtungen) mit Hilfe dieser Gestik. Lucentes [57] Arbeit fokussiert dagegen besonders auf die Steuerung eines interaktiven Graphiksystems über kamerabasierte Erfassung (und Spracheingabe). Die Auflösung (z.Bsp. der Zeigerichtung über den Kopf–Hand Vektor) und die Erkennungsrate ($\sim$ 10Hz) liegen aber in Bereichen, welche sie für den Einsatzzweck in immersiven Echtzeitsystemen als noch ungeeignet erscheinen lassen.

Symbolische Gesten

Symbolische Gesten sind solche Gesten, deren Bedeutung im sozialen und kulturellen Gebrauch eindeutig sind. Die einzelnen Zeichen der Taucher-sprache als auch die der verschiedenen Gehörlosensprachen[11] gehören in diese Kategorie. Symbolische Gestik fand schnell den Weg in die Benutzung als Mensch-Maschine-Kommunikationsmittel (z.Bsp. in [32]). Die Gründe dafür sind folgende: Um die Symbole einfach verständlich und interpretierbar zu machen, bestehen sie aus einfachen, klar zu unterscheidenden Gesten. Diese sind häufig deutlich durch eine eindeutige Postur definiert — ein Umstand, welcher auf einer technischen Ebene eine Erkennung dieser Symbole erheblich erleichtert. Auch ihre anschließende rein qualitative Interpretation ist eindeutig und kann in einer 1:1 Relation ohne Zuhilfenahme weiterer Informationsquellen (z.Bsp. multimodal) auf Systemfunktionen abgebildet werden. Die Art dieser Abbildung ist aber nur in den seltensten Fällen intuitiv und setzt ein Erlernen einer solchen Anwendungssprache voraus. Der Benutzer äußert beispielsweise eine vorher definierte Geste, das System führt anschließend die mit der Geste verknüpfte Operation aus. Weitere Modulationen der erwünschten Operation können entweder durch zusätzliche Gesten

[11] Hier wird nur Bezug auf die einzelnen Zeichen selbst genommen. Es gilt heute als unumstritten, dass die Gehörlosensprachen alle Aspekte einer natürlichen Sprache besitzen.

einer weiteren Extremität oder durch den verbalen Kanal erfolgen, wobei aber der intuitive sprachbegleitete Gebrauch dieser eindeutigen Gestik unwahrscheinlich erscheint. Beispiele des Gebrauchs symbolischer Gesten zur Steuerung computergraphischer Anwendungen sind zum Beispiel zu finden in [32]. Dennoch bieten symbolische Gesten eine einfache Lösung, um viele bisherige Interfaces durch eine gestische Schnittstelle zu erweitern und damit von den bisherigen Eingabegeräten zu lösen.

Beats

Die Klasse der sprachmarkierenden und sprachregulierenden Gesten erscheint — betrachtet vom kommunikativen Zweck aus — vordergründig weniger eindeutig als die bisher erläuterten Gestenklassen mit stark symbolischem Charakter oder mit starkem räumlichen Bezug. Der Begriff der Gestikulation verdeutlicht dieses zusätzlich. Es wird bei diesem Typus weniger von einer eindeutig zu identifizierenden Gestenart gesprochen, sondern vielmehr von Körperbewegungen, welche in Bezug auf ihre Verwendungen im kommunikativen Akt auf halbem Weg zwischen nicht-kommunikativer körperlicher Artikulation und einer gestischen Informationsübermittlung einzuordnen sind. Es ist daher schwer, diese Gestikulationen als intuitive Eingabemöglichkeit zu verwenden, da der kommunikative Gehalt viel eher im Verborgenen liegt als bei anderen Gestentypen. Ihr Gebrauch ist nur teilweise intentional, und wenn, dann dient er der Modulation des Kommunikationsaktes, nicht der Modulation des kommunizierten Faktums, womit es für die Interaktionssteuerung besonders nützlich wäre. Das Unterstreichen wichtiger Absätze mit dem Klopfen der Hand auf das Rednerpult demonstriert zum Beispiel einen intentionalen Gebrauch dieser Gestik. Viele Bewegungen fallen aber im Gegensatz dazu während der verbal/gestischen Informationsweitergabe rein intuitiv in eine enge Kopplung zwischen den beiden Modalitäten, womit eine bewusste Nutzung als reine Eingabequelle eher unnatürlich erscheint. Dennoch kann eine technische Auswertung dieser Gestikulationen im Zusammenhang mit der uni- und multimodalen Interpretation hilfreich sein. Welche Art von Querbezügen dabei Beachtung finden sollte, wird in Abschnitt 5.2 erläutert.

Von pantomimisch bis metaphorisch

Spricht der wenig intentionale kommunikative Gebrauch regulierender und markierender Gestikulation gegen einen primären Einsatz als Steuerung in einem Eingabesystem, so stellt sich der Sachverhalt bei pantomimischen und

metaphorischen Gestenexpressionen grundlegend anders dar. Sie kommunizieren eindeutig Informationen, nutzen dafür aber respektive entweder a) die gestische *Verbildlichung* eines mitunter sehr komplexen Handlungsablaufs oder b) eine Methode der metaphorischen Abbildung. Die Eigenschaften dieser verschiedenen Gestentypen betonen nicht die Modulation paralleler Äußerungen, sondern ihr Bedeutungsgehalt ist in hohem Maße entweder sehr abstrakt oder sehr komplex mit dem kommunizierten Faktum verknüpft.

Auch pantomimisch und metaphorisch sind keine ausschließenden Kriterien einer gestischen Klassifikation. Pantomimik impliziert zu einem gewissen Grad immer auch eine Abstraktion. Ob im Gebrauch einer pantomimischen Darstellung eines Vorgangs, etwa dem Öffnen einer Tür mit einem (hypothetischen) Schlüssel, oder dem metaphorischen Veranschaulichen, indem man mit Zeigefinger und Daumen eine kleine Spanne als Maß für den eigenen Gemütszustandes anzeigt: Die transportierte Information ist nur als Ergebnis einer abstrahierenden Abbildung des zu kommunizierenden Faktums zu verstehen. Ohne eine Vorschrift den Abbildungsvorgang umzukehren, steht der Sinngehalt nicht zur Verfügung. Einem Gebrauch solcher Gesten kann nicht der intentionale Charakter abgesprochen werden, die Schwierigkeiten liegen auf der Umsetzung in einem realen Interpretationssystem. Auf dessen Ebene müsste ein elaboriertes *Wissen oder Weltbild* über mögliche metaphorische Beziehungen vorhanden sein, um auf deduktive Weise solche Art von Äußerungen zu verstehen. Daher können sie momentan nur theoretisch in wenigen speziellen Einzelfällen als Eingabemöglichkeit in Betracht gezogen werden.

2.4.3 Methoden zur Gestenerkennung

Prinzipiell konnten drei verschiedene konzeptuelle Ansätze der technischen Realisierung einer Gestenerkennung identifiziert werden. Lernbasierte Methoden bieten nach ausreichender Trainingsmenge relativ hohe Erkennungsraten, setzen aber die Bereitstellung von genügend Beispieleingaben voraus. Betrachten wir den angestrebten Einsatz in einer VR-Anwendung, so müssen natürlich neben der Bereitstellung einer reinen Klassifikationsleistung — dem Feststellen, welche Geste geäußert wurde — auch die Möglichkeit der quantitativen Auswertung dieser geschaffen werden. Die im vorangegangenen Abschnitt identifizierten, für eine natürliche Interaktion besonders geeigneten Gesten sind ja eben nicht nur durch ihren Formtypus ausgezeichnet, sondern vermitteln durch die Abstraktion ihrer Expression im Raum gerade das kommunizierte Faktum. Hier reicht nicht die reine Identifikation der Geste, wie es beispielsweise bei den symbolischen Gesten möglich

ist, sondern es müssen auch analytische Möglichkeiten zur Verfügung gestellt werden, um den jeweiligen Kommunikationsgehalt auszuwerten. So werden z.Bsp. auch Funktionen benötigt, welche die Richtung der deiktischen Geste oder die Bahn und damit die Form der Trajektorie erkennen und beschreiben.

Die bisher für eine Notation gestischer Expression betrachteten Systeme — als Möglichkeit einer Kodierung auch nicht-symbolischer Gesten — leisten nicht die hier benötigte Parametrisierung. So lassen sich zwar begrenzt Formen notieren, aber deren Ausprägung zum Beispiel in der Richtung, in Form eines beschriebenen Kreisbogens oder der genauen Lage eines Rotationsmittelpunktes wird nicht gewährleistet. Betrachtet man diesen Sachverhalt unter dem Aspekt der Repräsentation gestischer Äußerung, so besitzen die bisher existierenden rein symbolischen Notationen für die sich hier stellende Aufgabe des Einsatzes auch mimetischer und ikonischer Gesten in der VR weder die nötige Granularität, noch findet eine Form der Parametrisierung Anwendung. Dagegen ist eine Repräsentation durch eine solche Notation durchaus angemessen, um das eigentliche Vorliegen einer Geste zu bezeichnen — eine Möglichkeit, welche durch eine rein analoge Kodierung schwerlich erbracht werden kann. Letztendlich wird ein hybride Repräsentation der Aufgabe angemessen. Symbolisch und zeitdiskret, um das Vorliegen gestischer Perzepte zu erkennen und daraufhin eine multimodale Integration zu ermöglichen, und analog, um das Erkennungsresultat auszuwerten und in der Anwendung zur Verfügung zu stellen.

Viele der auf Trainingsmethoden operierenden Erkennungssysteme zielen auf die Abbildung in eine sehr abstrakte, rein symbolische Repräsentation. Dagegen bieten universellere Verfahren, welche Lernmethoden erst einmal zur spezifischen Feststellung der Körperkonfigurationen einsetzen, eine sehr gute Ausgangsbasis, die damit ermittelten analogen Daten in einem weiteren Auswerteschritt sowohl zu klassifizieren als auch den analogen Gehalt auszuwerten. Da viele dieser Verfahren eine kamerabasierte visuelle Erkennung einsetzen, eröffnen sich damit Möglichkeiten kabelgebundene Sensorsysteme erfolgreich zu ersetzen und die Grundlage einer weiteren Gestenanalyse zu liefern. Angestrebte Ziele müssen in diesem Bereich zum einen eine Erhöhung der derzeit möglichen reinen Erkennungsgeschwindigkeit sein. Für eine kontinuierliche Analyse der Gestik kann ein unterer Grenzwert von mindestens 20Hz als maßgebend betrachtet werden. Zum anderen stellt eine feinere Granularisierung der erkannten Körperteile eine weitere Herausforderung dar. Bisherige Systeme, welche nur Hand- und Kopfpositionen ermitteln, bieten nur eine sehr grobe Auflösung. Bei der Auswertung zum Beispiel von deiktischen Gesten ist damit eine breite Streuung in der Analyse der Richtung begründet. Für eine manipulative Arbeit in einer Virtuellen Umgebung

mit vielen — auch sehr unterschiedlich großen Einzelobjekten — ist dieses nicht ausreichend.

2.4.4 Multimodale Eingaben in VR

Konzeptuelle Lösungen zur Einbettung multimodaler Eingabesysteme in VR-Anwendungen werden nur in wenigen Randbemerkungen beschrieben. Ansätze dazu, welche den geometrischen Eigenschaften der dargestellten Objekte gesondert Beachtung schenken, finden sich zum Beispiel in [87]. Viele der explorierten Systeme beschränken sich entweder auf einfache 2D-Graphiken oder sie verfolgen keinen Immersionsanspruch durch Headtracking und Stereoskopieverfahren. Die hier besonders kritischen Fragestellungen der Berücksichtigung temporaler Zusammenhänge werden nicht diskutiert. So ist es bei einer immersiven Betrachtung durchaus möglich, während sprach-gestischer Äußerungen den Benutzerstandpunkt, und damit die relative Sicht auf die Anordnung der Objekte zu verändern. Relative sprachliche Objektbezeichnungen sowie absolute gestische Deixisäußerungen finden kontinuierlich statt. Die damit in einer virtuellen Szene geäußerten sprach-gestischen Referenzen haben aber nur zum konkreten Äußerungszeitpunkt Gültigkeit. Eine Analyse der Eingaben wird aber immer zeitversetzt stattfinden. Sprach- sowie Gestenerkennung müssen genügend Informationen für eine vollständige Erkennung zur Verfügung haben und leisten eine erfolgreiche Segmentierung häufig nur durch ausreichende Informationen über das der aktuellen Äußerung nächst folgende Perzept. Dieses bedingt eine Latenzzeit in der Erkennung. In statischen Systemen ist ein solcher zeitlicher Versatz nicht kritisch. Durch die allgemein akzeptierte Lösung der Zeitstempelung der registrierten Perzepte kann eine spätere Rekonstruktion immer stattfinden. Im Falle immersiver VR bedeutet dieses aber eine Rückschau in die zu einem Äußerungszeitpunkt gültigen Szenenprojektion. Die Frage ist hier, eine geeignete Möglichkeit zu finden, diese Rückschau zu gewährleisten.

Eine Einbettung von Tastatur- und vor allem von Mauseingaben wird von allen VR-Systemen geleistet, da es sich hier quasi um einen Standard für die Interaktion mit Desktop-basierter VR handelt. Daneben gibt es ebenfalls Ansätze, die Ausgabegeräte und die Ausgabeart als dynamische Bestandteile in der Szenen- oder der Applikationsstruktur zu verankern. Wesentliche in der VR benötigte Interaktionen wurden bereits als sprach-gestisch realisierbar identifiziert. Um eine allgemeine Möglichkeit zu deren Auswertung zu gewährleisten, bedingt dies strukturelle Anpassungen bisheriger Konzeptionsmethoden von VR-Anwendungen. Die enge Verzahnung sprach-gestischer Äußerung mit den erfassbaren Objekteigenschaften und -attributen sowie

der spatialen Anordnung der Objekte in der Szene, benötigt für eine gelingende Interpretation Zugriff auf diese Informationen. Hier wird der Bedarf einer methodischen Erweiterung oder Reorganisation in der Modellierung dieser Objektstrukturen in VR-Anwendungen deutlich.

In den folgenden Kapiteln werden, aufbauend auf den Betrachtungen bisheriger Ansätze sprach-gestische Eingaben für computergraphische Anwendungen einzusetzen, neuartige Konzepte für eine multimodale Interaktion speziell mit VR-basierten Systemen entwickelt. Die Forschungsergebnisse zum kommunikativen Gebrauch von Gestik und den benötigten Interaktionen in VR-Anwendungen führen zu den in diesem Einsatzgebiet als natürlich und praktisch anzusehenden Gestentypen. Die Phänomenologie menschlicher Gestik in Bezug auf die spatiotemporale Gestenexpression sowie die bisherigen Verfahren zur maschinellen Gestenerkennung liefern die Grundlage, um im VR-Kontext eingebettete Erkennungsmethoden für bedeutsame Gestenmerkmale zu entwickeln. Eine Betrachtung von Beispielen verbaler Handlungsanweisungen ermitteln die für eine multimodale Auswertung wichtigen Integrationseinheiten einerseits auf sprachlicher und andererseits auf gestischer Seite. Resultierend werden weitgehend allgemeingültige Methoden eines Einsatzes multimodaler Interaktionen in Virtuellen Umgebungen ausgearbeitet. Der Fokus liegt auf dem Aspekt der systematischen Einbettung sprach-gestischer Steuerung in den Kontext existierender Ansätze zur Generierung von VR-Systemen. Sowohl die Bereitstellung der während der multimodalen Interpretation benötigten Informationen der virtuellen Szene, als auch die konzeptuelle Umsetzung der benötigten Aktionen in der Szene werden explizit ausgearbeitet. Die entwickelten Methoden erweitern dabei bisherige generelle und teils standardisierte VR-Modellierungsansätze um die benötigten Konstrukte zur Einbettung einer multimodalen Interaktion.

Kapitel 3

Basis der Interaktionsgestaltung

3.1 Mögliche Interaktions-Gestenformen

In Abschnitt 2.2 wurden die unterschiedlichen Gestentypen in Bezug auf ihre kommunikative Verwendung beschrieben. Die Spezifikation benötigter Interaktionen (s. Abschnitt 2.3.2) im Vergleich zur der von Nespoulous und Lecour eingeführten Typologie (s. Abschnitt 2.2.2) identifizierte die illustrativen Gesten als besonders geeignete Basis, um benötigte Informationen für Interaktion in 3D-basierten immersiven Anwendungen gestisch zu kommunizieren. In der hier verwendeten Fassung wird dabei der Begriff der deiktischen Gesten um zusätzliche Konzepte erweitert (s.u.). Zusammenfassend sollen nun die in der vorangegangenen Diskussion als Eingabemöglichkeit in Virtuellen Umgebungen und speziell der Virtuellen Konstruktion besonders nützlich erscheinenden Gestentypen rekapituliert werden.

Deiktische Gesten: Hier wird eine erweiterte Definition der Deixis über einfaches Zeigen hinaus (wie etwa in [74]) verwendet. Die Blick- und Körperausrichtung wird ebenfalls in Betracht gezogen, um gestische Informationen für eine Referenzanalyse — zur Objekt- und Ortsidentifikation — bereitzustellen.

Kinemimische/mimetische Gesten: Diese Gesten können entweder gewünschte Aktionen beschreiben oder eine bereits interaktiv stattfindende Interaktion modulieren, also bestimmte Parameter kontinuierlich verändern.

Weiterhin können bewegte Objekte durch die Nachahmung ihrer Bewegung identifiziert werden.

Ikonische Gesten: Diese Art von Gestik kann auf zweierlei Arten für Interaktionen ausgewertet werden:

- Spatiographic — um eine spatiale Konfiguration eines Objektes anzuzeigen und damit weitere Informationen für einer Referenzanalyse zu ermitteln bzw. eine gewünschte Konfiguration anzuzeigen.
- Pictomimic — um Änderungen von Formattributen, etwa die Länge oder Größe eines Objekts, gestisch anzuzeigen oder um Objekte zu beschreiben.

Es lassen sich im Rahmen einer Interaktionssteuerung natürlich auch andere Gestentypen auswerten. Sie dienen ebenso der Kommunikation, ihre Bedeutung ist aber in vielen Fällen sehr viel abstrakter. Die bisherigen Untersuchungen und Erfahrungen zum Gebrauch gestischer Expression im gewählten Szenario bestätigen die Verwendung dieser ausgewählten Gestiktypen. In dreidimensional präsentierten Umgebungen dienen primär die direkt spatial und dynamisch zu interpretierenden Gesten der intuitiven natürlichen Kommunikation.

Symbolische Gesten: Direkte Abbildung auf Systemoperationen.

Beats, sprachmarkierende Gesten: Segmentierung, technische Gestenerkennung.

Sowohl die symbolischen Gesten als auch die Beats nehmen eine Sonderstellung ein. Erstere, weil sie erlernt werden müssen aber den bisherigen Interfaces stark ähneln, letztere, weil sie bei einer technischen Umsetzung der Gestenerkennung und multimodalen Integration nützlich scheinen, jedoch für eine Interaktion nicht primär kommunikativ gebraucht werden. Sie sind im Prinzip ein Spezialfall der spatiotemporalen Gestenexpression im Kontext von coverbalen Gesten. Auf Grund ihrer explizit in der Literatur zu findenden Identifikation, werden sie bei dieser Aufstellung jedoch genannt. Die drei zuvor beschriebenen Gestentypen sowie symbolische Gesten werden im Folgenden kurz als *Interaktionsgestik* bezeichnet.

Durch die im CVK vorhandene natürlichsprachliche Texteingabekomponente wurden bereits Möglichkeiten geschaffen, um sprachlich Aktionen

auszulösen — etwa um Objekte zu instanziieren oder Verbindungen zu etablieren. Dennoch wurde auf eine mausbasierte Eingabe für eine direkte Manipulation nicht vollständig verzichtet. Es gibt offensichtlich Situationen, in denen die textbasierte sprachliche Eingabe nicht mehr ausreicht und der Anwender zu anderen "Mitteln" greift, um seine Interaktionswünsche auszudrücken. Im Fall des CVK, wie in fast allen CAD- und VR-Systemen, sind diese Mittel spezialisierte Werkzeuge zur direkten Manipulation der verschiedenen Primitiven mit Hilfe von zu benutzenden Eingabedevices. Die menschliche Gestik als ein weiteres primäres Kommunikationsmittel, wurde in diesem Bereich bisher wenig beachtet (s.a. Abschnitt 2.1). Als Eingabemodalität bietet Gestik aber einen prinzipiellen Vorteil in Situationen, in welchen die verbale Ausdrucksform nicht mehr ausreichend oder hochgradig ambig ist. Gestik ist inhärent räumlich verankert. Durch diese Eigenschaft allein lassen sich viele spatiale und metaphorisch spatiale Informationen direkt durch Gestik kommunizieren. Dieser Umstand lässt gestische Eingabe insbesondere für die Interaktion mit Virtuellen Umgebungen geeignet erscheinen. In Verbindung mit sprachgestützten Interaktionen ergibt sich ein erhebliches Manipulationspotential für die VR.

3.2 Diskrete und kontinuierliche Interaktionsabläufe

Nach den grundsätzlichen Betrachtungen der in Frage kommenden gestischen und sprachlichen Äußerungen wird auf die prinzipiellen Auswirkungen bei deren Umsetzung in einer VR-basierten Anwendung eingegangen. Bei der Betrachtung der möglichen multimodalen Interaktionen ergeben sich zwei unterschiedliche Interaktionsabläufe. Für den späteren Einsatzzweck in einer Virtuellen Umgebung müssen diese berücksichtigt werden. Sie bedingen prinzipiell verschiedene Modellierungen während der technischen Realisierung.

Die erste Methode einer Ablaufsteuerung benötigt im Vorfeld eine vollständige Spezifikation aller benötigter Parameter. Ist dieses gegeben, so ist eine Interpretation des Interaktionswunsches eindeutig. Der Interaktionswunsch — und damit die Veränderung der dargestellten Szene — kann in einem Schritt erfolgen. Ist beispielsweise ein Verbindungswunsch geäußert worden, müssen die beiden Verbindungspartner und Verbindungsorte bestimmt werden. So bietet die rein textbasierte Eingabe des CVK eine solche Interaktionsform an.

Sind Eingabeinformationen unterspezifiziert, so sind zwei Vorgehensweisen möglich. Die Interaktion kann abgebrochen werden, evtl. mit Rückfragen an den Benutzer, oder es kommt im CVK zum Einsatz von Heuristiken, welche die fehlenden Informationen liefern. Dieses geschieht unter Zuhilfenahme von Diskurswissen, um etwa Referenzen durch zuletzt bearbeitete Objekte aufzulösen, oder es werden sinnvolle Vorentscheidungen bei der Modellierung der Bauteile berücksichtigt: Fehlt im gegebenen Beispiel der Verbindungsort auf einem der Bauteile, und sind mehr als ein kompatibler Port vorhanden, so kann auf Defaultwerte für die Ports und *Hotspots* für die örtliche Lokalisation zurückgegriffen werden. In all diesen Beispielfällen lässt sich die Manipulation in einem Zuge durchführen. Weitere Eingaben — im Sinne einer Konkretisierung – sind von Benutzerseite aus nicht notwendig.

Eine zweite Interaktionsmethode wird bei der Betrachtung mausgestützter Interfaces deutlich. Das Eingabegerät (Inputdevice) dient als Medium, um interaktiv benötigte Informationen während der laufenden Manipulation schrittweise zu vermitteln. Die Aufgabe während des Interfacedesigns ist es, möglichst adäquate, intuitive Möglichkeiten zu schaffen, die Deviceinformationen in Manipulationen umzusetzen. Im CAD- und Grafikbereich ist es zum Beispiel de facto Standard, die räumliche Orientierung und Ausprägung der Objekte in verschiedenen Ansichten verändern zu können. Die Problematik in diesem Vorgehen liegt in der unterschiedlichen Dimensionalität des Eingabegerätes im Vergleich zu den veränderbaren Parametern. Dennoch hat sich die interaktive Manipulation als äußerst hilfreich erwiesen. Prinzipiell sind also die zwei folgenden unterschiedlichen Interaktionsabläufe zu unterscheiden:

1. Diskret: Die Interaktion erfolgt in einem Schritt.

2. Kontinuierlich: Die Interaktion erfolgt interaktiv schritthaltend.

Besonders der kontinuierlichen Interaktion wird während des folgenden konzeptuellen Systemdesigns Aufmerksamkeit geschenkt. Diese Interaktionsform ist in einer VR-Umgebung besonders geeignet, um exploratives Vorgehen zu ermöglichen, benötigt während der technischen Umsetzung aber eine prinzipielle Designmethode, ähnlich den Message-Loop basierten Programmstrukturen und Auswertungsmethoden in WIMP (Windows Icons Menu Pointing devices)-Systemen. Im Unterschied zu diesen kann aber eine gestische Analyse nicht auf diskreten Daten niedriger Dimensionalität arbeiten, etwa auf der planaren Bewegung eines Mausdevices. Die Basis für eine Gestenanalyse und -erkennung sind höherdimensionale Daten über die Körperkonfiguration, welche kontinuierlich mit Hilfe von Abstraktionsverfahren

über die spatiotemporalen Merkmale ausgewertet werden. Erst eine darauf aufbauende Abbildung des *Gestenraums* in den *Interaktionsraum* macht die Bewegungsinformationen für eine Umsetzung in eine Manipulation zugänglich. Im Folgenden werden die zwei verschiedenen Interaktionsabläufe noch einmal zusammengefasst:

Diskrete Interaktionen: Resultieren die uni- oder multimodalen Eingaben in einer vollständig beschriebenen Anweisung, so kann die Manipulation in einem Schritt erfolgen. Ist beispielsweise eine Rotation eines Objektes initiiert worden, sind weiterhin das Zielobjekt, die Rotationsachse, die Rotationsweite und -richtung bekannt, so kann die Szene in einem Schritt verändert werden. Dieses Vorgehen ist für viele im CVK implementierten Funktionen geeignet. Beispiele für diskrete Interaktionen sind die Instanziierung neuer Objekte, deren Auswahl, die Etablierung und Lösung von Verbindungen und mehr. Auch multimodale Eingaben implizieren — je nach Art der begleitenden Gestik — häufig eine diskrete Interaktion. Viele deiktische Bezüge und ikonische Gesten kommunizieren ein spezielles *atomares* Faktum, welches je eine benötigte Information in der semantischen Interpretation eines Interaktionsframes liefert. Ist dieser vollständig, so kann die Aktion ausgeführt werden.

Kontinuierliche Manipulationen: Einer der Vorzüge der Virtuellen Umgebungen ist die Möglichkeit der fortlaufenden Inspektion der Szene zu jedem Zeitpunkt während der Durchführung der Manipulationen. In Abschnitt 2.3.1 wurde bereits auf den Aspekt des explorativen Veränderns einer Szene hingewiesen. Bisher werden diese Form von Manipulationen mittels herkömmlicher Eingabegeräte zur Verfügung gestellt. So können im CVK mit Hilfe von Mauseingaben Objekte selektiert und anschließend manipuliert werden. Die hier beschriebenen Arbeiten zielen auf eine sprach-gestische Eingabe. Wie bereits gezeigt wurde, bietet mimetische Gestik das adäquate Eingabemedium, um sukzessive eine initiierte Aktion zu modulieren. Die Gestik beschreibt einen Vorgang, während der laufenden kontinuierlichen Interaktion wird die jeweilige Beschreibung schritthaltend interpretiert und angewendet. Dieses Vorgehen erlaubt die schrittweise Adaption der Aktion. Da sich gestische Eingaben im Gegensatz zum bisherigen Gebrauch spezifischer Eingabegeräte nicht eindeutig interpretieren lassen, wird eine Methodik entwickelt, um diese Interpretations- und Mittlerfunktion homogen in einer kontinuierlichen Manipulation einzubetten. Im folgenden Abschnitt wird diesem Umstand mit der Einführung spezieller "Vermittlerobjekte" in den Kontext der VR-Render-Loop Rechnung getragen.

3.3 Aufgabenstellung

In den Abschnitten 2.2.1 und 2.2.2 wurden — neben einer Beschreibung der spatiotemporalen Merkmale gestischer Äußerung — die Forschungsbestrebungen geschildert, die Gestik in Bezug auf ihren kommunikativen Gebrauch zu klassifizieren. Auf dieser Grundlage ist es möglich, zwei unterschiedliche Extrema der gestischen Informationsweitergabe und ihrer Interpretation zu differenzieren. Die hier erfolgende Unterscheidung betrachtet **a)** einen rein **qualitativen** Aspekt gestischer Information — also die Frage, wann welche Geste geäußert wurde — im Gegensatz zu **b)** deren **quantitativen** spatiotemporalen Aspekten, der Frage nach der Ausprägung der verschiedenen die Geste ausmachenden Form- und Bewegungsmerkmale.

Bei der ersten Fragestellung interessiert allein das Faktum des Vorliegens der Geste anhand der Erkennung bestimmter Merkmale. Im einfachsten Fall handelt es sich hier um eine einfache Posturerkennung, kann aber ebenso die Erkennung einer (dynamischen) Geste, etwa eines Winkens, bedeuten. Wichtig ist, dass als Resultat allein das Vorliegen der Geste als qualitative Information ermittelt wird.

Dieses impliziert eine generelle Sichtweise auf Gestik und deren sprachähnlichem Gebrauch und ignoriert dabei die primären, durch ihre Ausführung gegebenen spatiotemporalen Gestikmerkmale. Viele Anwendungen zur Gestenerkennung konzentrieren sich allein auf die Klassifikationsfunktion — so zum Beispiel bei der Erkennung symbolischer Gesten oder von Elementen der Zeichensprachen. Hierbei spielt es kaum eine Rolle wie expressiv die Geste ausgeführt wird, allein die Tatsache, dass sie ausgeführt wird, ihre Benutzung ist von Relevanz. Selbst der metaphorische oder ikonische Gebrauch von Gesten steht teilweise stellvertretend für einen konkreten Begriff. Das Beispiel des Fingerdrehens in der Luft zur Beschreibung des eigenen Gemütszustandes veranschaulicht dieses plastisch.

Die Vielschichtigkeit menschlicher Gestik geht allerdings über diese rein qualitativen Aspekte weit hinaus. Das Erkennen von Beats beispielsweise lässt sich zwar als Event symbolisch notieren, die Dynamik kommuniziert aber zusätzliche Information, etwa die Stärke eines Gemütszustandes, welche bei einer solchen Notation ignoriert würde.

Je mehr wir uns der Interaktionsgestik zuwenden, um so wichtiger wird der quantitative Aspekt der spatiotemporalen Expression. Die geometrischen Eigenschaften der beschriebenen Bahnen und die dynamischen Merkmale sind die Informationen, welche im laufenden Diskurs entweder direkt oder metaphorisch interpretiert werden. Besonders im direkten Fall prädestiniert dieses eine gestische Kommunikation als Interaktionsmittel in Virtu-

ellen Umgebungen. Die Art der vermittelten Information ist hier weniger als abstraktes Symbol zu sehen, sie ist explizit in der Ausprägung der spatialen und dynamischen Äußerung vorhanden. Eine Analyse von Attributen wie Richtung, Form, Weite oder Geschwindigkeit kann nicht auf einer rein qualitativen Ebene erfolgen. Sie muss auf Basis eines quantitativen nummerischen Modells eine Abbildung von geometrischen und temporalen Beziehungen auf noch zu definierende Merkmalsbegriffe bereitstellen, jedoch gleichzeitig den Zugriff auf die nummerische Basis erlauben. Die dort verankerten Werte dienen so, je nach Kontext, der Umsetzung in eine Interaktion.

Neben der sich stellenden Aufgabe, Gesten zu erkennen und auszuwerten, besteht die Anforderung, Methoden für eine multimodale Integration zu entwickeln. Zu diesem Zweck müssen in einem ersten Schritt die in Frage kommenden sprachlichen Äußerungen identifiziert und analysiert werden. Daraufhin müssen Zusammenhänge sprach-gestischer Äußerungen herausgearbeitet werden. Die Art dieser Zusammenhänge stellt die Frage nach einem geeigneten Formalismus für deren Repräsentation. Auf der Basis eines solchen Repräsentationsformalismusses müssen die multimodalen Eingaben zueinander in Bezug gesetzt und im Hinblick auf eine mögliche Interaktion technisch interpretiert werden.

Auf der Seite der VR und den bisher dort eingesetzten Eingabemethoden ist es die Aufgabe, neue Interaktionsmetaphern auf Basis multimodaler Eingabemöglichkeiten zu schaffen. Zur Abstraktion von individuell unterschiedlichen Sensoriken und zur Generierung von für die Gestenauswertung benötigten Basisinformationen müssen geeignete Verfahren ausgearbeitet werden. Dieses soll unter Berücksichtigung vorhandener Ansätze geschehen. Es sollen neue Konzepte zur Erweiterung struktureller Komponenten entwickelt werden, so etwa die Einführung neuer Knotenklassen zur Einbettung in einen Szenengraphen oder die Anreicherung repräsentierter Objekte einer virtuellen Szene durch zusätzliche Attributbeschreibungen. Zum anderen müssen operative Modelle und funktionale Fähigkeiten auf diesen zuvor entworfenen statischen Konzepten bereitgestellt werden. Diese Erweiterungen sind in die VR-Ablaufstrukturen einzubetten, ohne die in der VR geltenden kritischen Randbedingungen der Echtzeitfähigkeit zu verletzen.

Kapitel 4

Ein allgemeiner Ansatz zur Gestenerkennung in der VR

4.1 Gestenmerkmale und ihre Verankerung

4.1.1 Bezugssysteme der Interaktionsgestik

Die Betrachtung des Gebrauchs von Gestik als Kommunikationsmittel erfolgte zur Identifikation derjenigen Gestentypen, die für einen Einsatz in interaktiven Systemen als besonders nützlich erscheinen. Mit der damit einhergehenden Begriffsbildung wurde eine gemeinsame Ontologie der untersuchten Domäne eingeführt. Die Konzeption eines Systems zur Gestenerkennung und -analyse erfordert aber neben der Feststellung, *was* analysiert werden soll, ebenso eine Aufstellung von bezeichnenden phänomenologischen Merkmalen, welche eine Erkennung ermöglichen. Um hier eine Basis für ein automatisches Analyseverfahren zu schaffen, fehlt eine genaue Aufstellung der spatiotemporalen Äußerungseigenschaften menschlicher Gestik. Nach der hier getroffenen Definition (s. Abschnitt 2.2.1) ist eine Geste in ihrem äußerlichen Erscheinungsbild eine dynamische Abfolge von zusammenhängenden Konfigurationen des Bewegungsapparates. Dabei sind aber durchaus nicht alle Körperpunkte von gleicher Wichtigkeit, sondern es ist in vielen Fällen ausreichend, nur einen Teilbereich des Körpers für die Rezipierung und erfolgreiche Interpretation gestischer Information zu betrachten.

Die für die Interaktion als nützlich angesehenen Gesten werden dabei vor allen Dingen durch die oberen Extremitäten ausgeführt. Allein die Deixis sieht den gesamten Bewegungsapparat als Referenzsystem, wird aber ebenfalls in besonderem Maße durch die Arme, Hände und den Kopf — als Blick-

richtungsapproximation — ausgedrückt. Generell sind die Arme und Hände primäre Vermittlerorgane menschlicher Gestik, vor allem bei den für die Interaktion als besonders nützlich identifizierten Gestentypen. Hände und Arme bilden dabei eine Einheit, welche durchaus gleichzeitig unterschiedliche Informationen vermitteln kann. Die Hände, betrachtet als Endeffektoren der Arme, sind besonders hilfreich, um gestische Expression feiner zu granularisieren und auszudrücken. So kann in einer gestischen Bewegung die Armtranslation die Länge eines Körpers vermitteln, die Finger der entsprechenden Hand aber gleichzeitig die Form oder Oberfläche des Objektes anzeigen. Weiterhin wird, vor allem zu beobachten bei der ikonischen Gestik, zweiarmig gestikuliert. Dieses geschieht fast ausschließlich symmetrisch, allerdings müssen die *Symmetrieverläufe* dabei nicht immer den Symmetrien des menschlichen Körperbaus entsprechen. Bei ikonischen Gesten mit starkem deiktischem Anteil und Bezug auf ein externes Referenzsystem, wie es beispielsweise bei der Formbeschreibung eines im umgebenden Raum gelagerten Körpers vorliegt, wird häufig ein von der Körperlage vordefiniertes Referenzsystem adaptiert. Mehr dazu wird im noch folgenden Abschnitt 4.1.2 erläutert.

Für die Erfassung der Interaktionsgestik ist es notwendig und in den meisten Fällen ausreichend, sowohl die Position und Orientierung der Hände bzw. der Handwurzeln als auch die Handformen selber zu analysieren. Letztere sind geprägt durch die einzelnen Fingerstellungen der Hände, deren Streckung und Spreizung.

Um die Blickrichtung zur Deixisauswertung zu betrachten wäre es idealerweise mindestens notwendig, die Kopf- und die Augenrichtung zu berücksichtigen. An dieser Stelle soll dabei auf einen wichtigen Zusammenhang zwischen Augenrichtung und Kopfstellung hingewiesen werden, da dieser bei einer späteren technischen Umsetzung hilfreich ist. Das menschliche Auge hat ein unterschiedliches Auflösungsvermögen. Am schärfsten sehen Menschen, mit recht engem 10° Sichtfeld, im fovealen Bereich. Bei einer Blickrichtungsänderung können aber die Augen sowohl in horizontaler als auch in vertikaler Richtung um 60° Halbsphärenwinkel bewegt werden [30], um andere Bereiche im fovealen Bereich zu fokussieren. Augen- und Kopfrichtung sind dabei eng gekoppelt. Wird bei diesen ausgeführten Fokussierungsbewegungen für einen kurzen Zeitraum eine Auslenkung der Augenposition innerhalb der Bewegungsspielräume eingenommen, so wird der Kopf des Benutzers nachgeführt, um die Augenposition in die "Ruhelage" zurückzusetzen. Dieses gilt in besonderem Maße, wenn die anvisierten Objekte und Orte in Bezug zum Betrachter weit verstreut liegen. Dieser Umstand gestattet es, in solchen Szenarien die Blickrichtung über die Kopfausrichtung zu approximieren.

Zusammenfassend sind für eine automatische Erkennung und Auswertung der Interaktionsgestik Basisinformationen über folgende Körperextremitäten besonders wichtig:

- Form der Hände, Fingerstellungen
- Position/Ausrichtung der Arme und Hände (Handrücken)
- Position/Ausrichtung des Kopfes

Für die Betrachtung der Positions- und Ausrichtungsinformationen werden daher entsprechende rechtwinklige Referenzkoordinatensystem mit definiten Achsausrichtungen eingeführt und wie in Abbildung 4.1 folgendermaßen verankert:

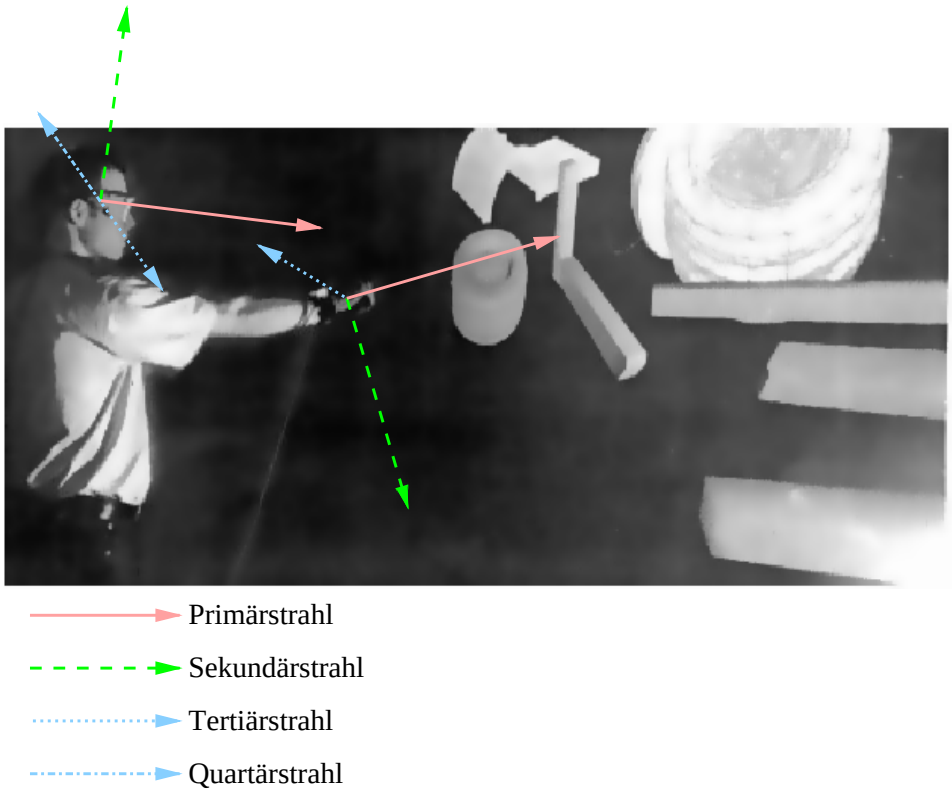

Abbildung 4.1: Verankerung der Referenzstrahlen auf einer Hand und am Kopf.

Lage und Strahlsignifikanz der Kopfrichtung

Das Kopfreferenzsystem wird im Kopfmittelpunkt auf Höhe der Augen verankert. Der *Primärstrahl* ist der orthogonal auf dem Gesichtsfeld stehende und vom Körper fortgehende Strahl. Der *Sekundärstrahl* steht senkrecht auf der Kopfoberseite und der *Tertiärstrahl* verläuft parallel zur Ohr-Ohr Linie und verlässt den Körper in Blickrichtung nach rechts. Durch die gegebene Kopfsymmetrie wird zu letzterem ebenfalls ein paralleler aber invers gerichteter *Quartärstrahl* verankert.

Lage und Strahlsignifikanz der Hand

Die Handreferenzsysteme werden auf den Handflächen verankert. Der *Primärstrahl* der Hand verläuft in der Verlängerung Ellenbogen-Handwurzel-Zeigefinger(gestreckt). Der *Sekundärstrahl* steht, bei flacher Hand, orthogonal auf der Handinnenfläche. Der *Tertiärstrahl* verläuft grob in Richtung abgewinkelter Daumen und steht dabei senkrecht zu den anderen Strahlen.

Diese Strahlklassifizierung entspricht den im gestischen Gebrauch auftretenden Bezugssystemen und wird bei der Diskussion über interne Symmetrien als spatiotemporale Gestenmerkmale Bedeutung gewinnen. Die Ermittlung dieser Ausgangsdaten ist als notwendig für ein Interaktionsgestik-Analysesystem anzusehen. Besonders zu beachten sind die durch die Strahlverankerungen gegebenen neuen Referenzsysteme. Deren Position und Ausrichtung werden als hinreichende Informationsquelle zur Verfügung gestellt und in einem gemeinsamen euklidischen Basissystem verankert. Dieses erlaubt konkrete nummerische Auswertung körperrelativer Bewegungen durch die Bereitstellung eines Abstandsmaßes.

4.1.2 Spatiotemporale Merkmale der Interaktionsgestik

Im Hinblick auf eine automatische Analyse müssen die bedeutsamen gestischen Bewegungen vor dem Hintergrund *nicht-kommunikativer* Bewegungen als solche identifiziert werden. Das Problem, die bedeutungstragenden Bewegungsphasen zu extrahieren, wird im allgemeinen als das *Segmentierungsproblem* bezeichnet. Dazu müssen die spatiotemporalen Bewegungsmerkmale detektiert werden und ferner Muster in der zeitlichen Korrellation von Merkmalen erkannt werden. Eine Form eines solchen merkmalsbasierten Segmentierungscues liefert beispielsweise das Handspannungsmodell nach [31]. Andere Merkmale wurden in [45] bei der Segmentierung der Gestlets sowohl durch die Erkennung von Bewegungsstops als auch durch die Analyse geometrischer spatialer Änlichkeitsbeziehungen ausgewertet.

Basierend auf diesen Vorbetrachtungen und durch die Inspektion experimentell ermittelter Daten (z.B. im speziellen VK-Kontext [94]) werden im folgenden acht spatiotemporale Merkmale vorgeschlagen, welche für die Trennung gestischer Artikulation als besonders wichtig erscheinen. Diese Zusammenfassung gründet auf der Betrachtung rein gestischer Äußerungen unter Ausblendung verbaler Informationen.

1. Aktion und Pause
2. Auslenkung aus Ruhestellung
3. Definite Posturform
4. Primitives Bewegungsprofil
5. Wiederholung
6. Interne Symmetrie
7. Externe Symmetrie
8. Externe Referenz

Betrachtet man diese Merkmale als bezeichnend für gestische Expression, kann der Wechsel zwischen Mustern aus diesen Merkmalen als Wechsel zwischen gestischen Äußerungen — als Segmentierung — interpretiert werden. Die Merkmale liegen jeweils für eine konkrete Geste in einem spezifischen temporalen Muster vor. Über die Art dieser Ordnung wird auf dieser konzeptionellen Ebene keine Annahme gemacht. Sowohl einfache sequentielle *Abfolgen* als auch sehr viel komplexere temporale Muster — etwa Überlappungen und Inklusionen — von Merkmalen können vorliegen. Viele der spatiotemporalen Merkmale machen bereits im einfachen singulären Fall den zeitlichen Verlauf als Kriterium explizit (Aktion und Pause oder Wiederholung). Die Merkmalskategorien sind hier nicht vollständig disjunkt. Sie resultieren aus dem Verlauf der Trajektorien und sie explizieren gewisse Regularitäten in den rein spatialen und den dynamischen Aspekten einer Bewegung. Die folgenden Ausführungen sollen die beschriebenen Merkmale näher erläutern.

Aktion und Pause

Ein grundlegendes Unterscheidungsmerkmal trennt die Phasen der körperlichen Ruhe von solchen mit Bewegungsanteil. Eine reine Abwechslung zwischen diesen Phasen ist kein eindeutiges Zeichen für gestische Kommunikation. Ein vorliegender Übergang wird erst dann als bedeutend für eine gestische Äußerung erkannt, wenn er nicht in der aktuellen Bewegung antizipiert wird.

Auslenkung aus Ruhestellung

Eine Konfigurationsänderung, welche zu einer Auslenkung aus einer definiten Ruhestellung oder Ruhekonfiguration führt, wird als "bemerkenswerte" Bewegung erkannt. Diese Ruhekonfiguration kann durchaus dynamische Aspekte haben — man denke an Gesten während des Voranschreitens. Das schon erläuterte Handspannungsmodell basiert beispielsweise auf einer Auslenkungsbetrachtung. Es ist ein Spezialfall der gemessenen Abweichung einer aktuellen Körperteilkonfiguration von einer anzunehmenden Ruhekonfiguration. Nach Beobachtungen wird ein Großteil kommunikativer Gestik der Hände vor dem Körper in Höhe des Brustkorbs ausgeführt (vgl. [42]). Diese Beobachtung lässt sich noch nicht wirklich als definite Posturform (s. nächster Absatz) klassifizieren, sondern sie ist als Merkmal zwischen einer allgemeinen Auslenkung und einer definiten eingenommenen Postur einzuordnen.

Definite Posturform

Viele Gesten, vor allem aus dem symbolischen Bereich, sind durch deutlich ausgeprägte Formmerkmale, vor allem der Hände, ausgezeichnet: "Daumen hoch", der gestreckt aufwärts gerichtete Daumen als OK-Zeichen oder der ausgestreckte Zeigefinger beim deiktischen Bedeuten. Diese reinen Formmerkmale bestätigen das Vorliegen der Geste. Dabei impliziert diese definite Form besonders eine Auslenkung aus einer Ruhestellung, sie ist ein Spezialfall einer solchen Auslenkung, in welchem die eingenommene Auslenkungskonfiguration besonderen symbolischen Charakter erhält.

Primitives Bewegungsprofil

Dynamische Bewegungen auf unterster Ebene bilden *primitve* Profile aus. Morasso [69] identifiziert diese bereits sowohl in den beschriebenen geometrischen Bahnen (s. *Krümmungsbetrachtung*) als auch bei der Betrachtung

der Dynamik der Trajektorien (s. *Glockenprofil* der Bewegungsgeschwindigkeit). Ein bedeutungsvolles Profil wird durch eine **Konstanz** eines spatialen oder dynamischen Parameters ausgeprägt. In spatialer Ausprägung entspricht dieses zum Beispiel der beschriebenen Krümmungsbetrachtung oder dem Aufeinanderfolgen von Bewegungsbahnstücken mit ähnlichen abstrakt-geometrischen Eigenschaften. Dieses können sowohl einfache Kolinearitäten von Bahnstücken zueinander, zu durchschrittenen geometrischen Primitiven wie Ebenen, aber auch sukzessiv eingehaltene Winkel oder Winkeländerungen sein. In dynamischer Ausprägung sind besonders Konstanz in der Bewegungsgeschwindigkeit rezipierbare Merkmale einer Bewegung.

Wiederholung

Formgleiche gestische Äußerungen können mikro- und makroskopisch in festen zeitlichen Muster mehrmals ausgeführt werden. Eine Gesamtgeste kann als Ganzes nach einer Pause wiederholt werden (makroskopisch) oder ein Form-Teilbereich wird innerhalb einer Geste repetiert (mikroskopisch), wie bei der Veranschaulichung einer Schlangenbewegung. Das Wiederholungsmuster kann dabei durchaus verschiedenen, auch rekursiven oder (rekursiv) rhythmischen Charakter besitzen.

Interne Symmetrie

Interne Symmetrien sind solche, bei denen entweder die impliziten Referenzsysteme mehrerer Extremitäten oder die Referenzsysteme, der durch sie vermittelten geometrischen Primitive, Symmetrien aufweisen. Bei der Verankerung der Referenzsysteme auf den Handflächen wurde die bestehende äußere Form der Hand zugrundegelegt. Dieses resultiert in einer Ausrichtung der Referenzstrahlen an den durch die Handflächen definierten Ebenen sowie den durch die Finger gegebenen Richtungen. Einfache interne Symmetrien emergieren zum Beispiel, indem dermaßen verankerte implizite Strahlen in eine statische kolineare Konfiguration gebracht oder kolinear zueinander bewegt werden. Komplexere Symmetrien beziehen sich auch auf gleichzeitige konstante spatiale Winkeländerungen und Ähnliches.

Externe Symmetrie

Externe Symmetrien sind solche, bei denen entweder die Referenzsysteme der beobachteten Extremitäten (s. Abschnitt 4.1.1) oder die Basisattribute vermittelter geometrischer Primitive, zum Beispiel bestimmter Strahl- oder

Achslagen, in symmetrischen Bezug zu körperfremden und extern verankerten Referenzsystemen sind. Die Symmetrien entstehen hier auf Grund der bewusstgemachten Ähnlichkeit der durch die Extremitäten dargestellten Formprimitive mit Formeigenschaften von externen Diskursreferenten.

Externe Referenz

Die impliziten Referenzstrahlen der benutzten Extremitäten spannen als Ebenennormalen imaginär den Körper umgebenden Raum in Teilräume auf. Sie definieren dadurch eine *Ordnung* der in diesem Raum befindlichen Objekte relativ zu diesen Ebenen und den Referenzstrahlen. Diese Ordnung ist sowohl gegeben durch die Nähe der Objekte zu den so erscheinenden Richtungsloten als auch durch den Abstand zu den dadurch gegebenen Ebenen.

Bei den extern verankerten Merkmalen der Symmetrie und Referenz genügt eine reine Rezipierung der Körperkonfiguration nicht mehr. Diese Form der spatiotemporalen Merkmale kann nicht ohne das externe Referenzsystem explizit gemacht werden. Die kommunikative Funktion liegt genau in der Vermittlung der mit diesem Abbildungsvorgang einhergehenden Referenzen und Symmetrien. Die mit der Geste übermittelten Informationen basieren explizit auf der Relation zu anderen im gemeinsamen Referenzraum befindlichen Entitäten.

Diese spatiotemporalen Merkmale gestischer Äußerung spezifizieren die Anforderungen an das hier entwickelte System zur Gestenerkennung. Auf ihrer Basis müssen Merkmalsdetektoren entwickelt werden. Die Aufgabe besteht auf dieser Ebene in der konzeptuellen Bereitstellung von Verfahren, um diese Merkmale aus gegebenen Rohdaten analytisch zu ermitteln und anschließend bedeutungstragende Information aus diesen Daten zu extrahieren. Darauf aufbauend müssen die zeitlich verorteten Merkmale zur Erkennung entsprechender Gestenmuster logisch verknüpft werden. Die folgenden Abschnitte gehen dabei speziell auf die Anforderungen an ein solches Erkennungssystem im Kontext einer VR-basierten Anwendung ein. Vorher wird eine kurze Erläuterung der in den hier geschilderten Arbeiten betrachteten und in der späteren Implementation verwendeten Sensoriken gegeben.

4.1.3 Sensorik

Die Gestenerkennung und -analyse basiert auf Daten über die Konfigurationen der sensorisch erfassten Extremitäten gemäß den gewählten Bezugs-

systemen nach Abschnitt 4.1.1. Für diese Aufgabe gibt es prinzipiell unterschiedliche Verfahren (s. Abschnitt 2.3). Die eingesetzte Sensorik muss für diesen Einsatzzweck mindestens die in Abschnitt 4.1.1 identifizierten Eingabedaten liefern. Um dynamische Aspekte der Gestik zu erfassen, muss eine minimale *Samplingrate* zur Verfügung gestellt werden. Ein solcher Grenzwert ist prinzipiell gegeben durch die maximale biologische Bewegungsgeschwindigkeit der erfassten Körperteile, eine durch die Muskelkontraktionsfähigkeit bestimmte Konstante. Dieses ist allerdings eine Maximalforderung, je nach Anwendungsfall kann die Rate auch deutlich niedriger liegen. Die Muskelkontraktion bestimmt die entsprechende Winkelgeschwindigkeit des bewegten Gelenks. Die resultierenden sensorischen Daten eines Endeffektors im euklidischen Raum basieren daher auf dem Sensorbefestigungspunkt und der entsprechenden Linklänge. Je nach Gestik werden so beispielsweise Handgeschwindigkeiten von deutlich über 1m/sec erreicht. Ein sinnvoller Minimalwert zur Auswertung dieser Art von Gestik liegt daher bei 20Hz, um hinreichend genaue Stützpunkte — mit einem räumlichen Auflösungsvermögen von mindestens 0.05m — über die beschriebene Bahn zu erhalten. Unter diesen Voraussetzungen können generell alle in Abschnitt 2.3 beschriebenen Sensorarten Verwendung finden.

In dieser Arbeit werden primär die in der Virtuellen Realität häufig zu findenden elektromagnetischen Positionssensoren und *Cyber*Gloves eingesetzt. Erstere bestimmen die relative Lage und Ausrichtung eines Sensors zu einem Referenzsystem anhand der durch ein starkes Magnetfeld induzierten Ströme in drei orthogonal in diesem Sensor befestigten Spulen. Letztere messen Gelenkbeugungen an den Fingern über Widerstandsänderungen in über den Gelenken angebrachten Bimetallstreifen. Die Positionssensoren befestigt man an den zu trackenden Körperstellen. In den hier vorgestellten Arbeiten liegen diese Punkte vorzugsweise, soweit dieses möglich ist, auf den bereits für die Interaktionsgestik als bedeutungsvoll identifizierten Punkten auf den Extremitäten (etwa den Handrückenmittelpunkten). Durch die Verwendung der in der VR verbreiteten, und als Quasistandard zu sehenden Sensorik, lassen sich die hier ausgearbeiteten Methoden auch auf Basis dieser Rohdatenverarbeitung auf andere Systeme übertragen.

Die Vorteile der eingesetzten Sensorik liegen sowohl im statischen spatialen Auflösungsvermögen als auch in der erreichbaren maximalen Samplingrate; hier sind je nach Konfiguration 100Hz und mehr zu erlangen. Weiterhin ergeben sich bei dieser Sensorik keine Verdeckungsproblematiken. Als nachteilig müssen der nach wie vor vorhandene Zwang zur zumindest teilweisen "Verkabelung" — mit den daraus entstehenden Unannehmlichkeiten während der Benutzung — sowie die Einschränkungen, welche mit dem elek-

tromagnetischen Prinzip verbunden sind, angesehen werden. Hier kommt es zum einen durch eingebrachte Metallgegenstände zu einer Messverzerrung in den Sensoren und somit zu Schwankungen in den Messungen, zum anderen resultieren die bei diesen Verfahren eingesetzten dynamischen Filterstufen in Unstetigkeiten in der relativen Auflösung.

Neben den beschriebenen eingesetzten sensorischen Verfahren wurden bereits Bestrebungen zur Entkoppelung von kabelgebundenen Devices unternommen. Im Rahmen der Kooperationbestrebungen fand eine erste prototypische Installation eines auf neuronalen Erkennungsalgorithmen basierenden optischen Erkennungssystems für die Ausrichtungsbestimmung der Arme statt [77]. Diese Aufgabe stellt unter den Illuminationsbedingungen einer Großleinwandprojektion besondere Anforderungen. Die Lichtverhältnisse sind, je nach Szene, extrem wechselhaft und in den meisten Fällen als im sichtbaren Lichtspektrum zu dunkel zu klassifizieren. Als Lösung wurde in diesem Projekt auf die Verwendung von externen Infrarotlampen und entsprechenden Kameras zurückgegriffen. Aktuelle weiterführende Arbeiten im Projekt GREFIT konzentrieren sich ebenfalls auf die optische Erkennung der Handposturen [75][76][78]. In der Kombination beider wäre eine völlige Entkoppelung von kabelbasierten Eingabedevices möglich. Als Nachteile dieser Verfahren müssen zum jetzigen Zeitpunkt besonders zwei Merkmale angesehen werden. Wie bei jeder extern verankerten, "beobachtender" Sensorik gibt es das Verdeckungsproblem. Es kann zu einer Körperstellung des Benutzers in der Interaktion kommen, in welcher getrackte Extremitäten durch den eigenen oder evtl. andere Körper verdeckt werden. Weiterhin ist die Samplingrate, bedingt durch die Vorverarbeitung der Bilder, deutlich geringer als bei den elektromagnetischen und elektrischen Verfahren. Auf einem zu diesem Zeitpunkt als Standard-PC (PentiumII 300) anzusehenden Rechner liegt sie je nach Teilaufgabe zwischen 3-5Hz (Lokalisation der Extremitäten) und $\tilde{1}$0Hz (Erkennung der Fingerstellung), einem Wert der durch entsprechende Hardware natürlich schon sehr nah am Einsatzminimum liegt.

4.1.4 Anforderungen an das Gestenerkennungs- und Analysesystem

Auf Basis der sensorischen Eingabedaten müssen die einzelnen spatiotemporalen Merkmale detektiert und diese zeitlich eingeordnet werden. Es werden Methoden benötigt, um die charakteristischen Merkmalsmuster (Templates) zu erkennender Gesten im Vorfeld zu definieren. Diese Templates müssen im laufenden System mit den aktuell präsenten Merkmalen in einem Matchingprozess verglichen werden. Die Vergleichsresultate sollen daraufhin bestimmte *Aktionen* auslösen.

Die beschriebenen Funktionen müssen im Kontext einer VR-Anwendung ermöglicht werden. Die dabei getroffenen Modellierungsentscheidungen sollen den Randbedingungen des Einsatzes in einem interaktiven Echtzeitsystem im Hinblick auf Latenzzeit und Berechnungsdauer genügen sowie eine weitgehend modulare Schnittstelle zu bestehenden Modellierungsansätzen für VR-Systeme bieten.

Nummerisch/symbolische Erkennung und Analyse

Betrachten wir die deskriptiven Definitionen (s. Abschnitt 4.1.2) der einzelnen spatiotemporalen Gestenmerkmale, so müssen Möglichkeiten geschaffen werden, um a) diese Merkmale aus den sensorischen Rohdaten zu gewinnen und b) diese einzelnen definitorischen Merkmale in kombinatorischer Art zu verknüpfen, um entsprechende Merkmalsmuster (s.a. Abschnitt 2.2.1) zu detektieren.

Dazu werden zuerst Verfahren bereitgestellt, um eine Abbildung von Berechnungsergebnissen auf den nummerischen Rohdaten hin zu einer symbolischen Beschreibung zu ermöglichen. Die Verknüpfungen dieser Symbole zu Regelbeschreibungen benötigt entsprechende Logik- und Vergleichsoperatoren. Diese Operationen auf der symbolischen Vergleichsebene werden dabei von nahezu allen bekannten Programmiersprachen erbracht. Um hier ein offenes Rapid-Prototype Vorgehen zu ermöglichen, werden die zu entwickelnden Konstrukte auf dieser Ebene mit Hilfe einer geskripteten Sprache einfach parametrisierbar gestaltet. Durch die Form einer so geschaffenen top-down Beschreibung der Definitionen wird dabei auch die Möglichkeit geschaffen, diese Deskriptionen zu überprüfen und ggfs. *interaktiv* zu modifizieren.

Wie aus den Deskriptionen der spatiotemporalen Merkmale ersichtlich ist, sind diese auf unterer Ebene durch nummerische Verfahren definierbar. Aktion oder Symmetrie sind deutliche Beispiele für eng an den nummerischen Daten orientierten Merkmalen. Letztendlich findet Gestik im euklidischen Raum statt, ebenso wie die Operationen im Zielsystem einer VR-Anwendung. Um die Auswertung von Bewegungen, geometrischen Constraints und kombinatorischen Formen nummerisch durchzuführen, eignen sich in erster Linie Methoden der analytischen Geometrie mit Hilfe linearer Algebra. Daneben existieren interessante alternative Verfahren basierend auf *Fitting*algorithmen oder der Betrachtung von spatialen *Ähnlichkeitsbeziehungen* in höherdimensionalen Abbildungen der Bewegungsdaten (*spatial aggregation* [4]).

Diese Berechnungsverfahren dienen im Folgenden als Grundlage, um gestische Expression im Kontext der VR-Anwendung zu analysieren. Mit den

hier beschriebenen Verfahren werden neben den kombinatorischen auch die nummerischen Aspekte der Erkennung, der Analyse und der Auswertung zur Verfügung gestellt. Dieses erfolgt auf high-level Basis mittels Intervallabschätzungen und Tests von auf low-level Basis nummerisch ermittelten Attributen.

Unabhängigkeit von Datenformaten und -raten

Ohne Beschränkung der Allgemeinheit sollen als sensorische Eingabedaten die in Abschnitt 2.3 definierten Basisinfomationen dienen. Wie später gezeigt wird, ist diese Vereinbarung keine grundlegende Einschränkung für das vorgestellte System, sondern wird in den folgenden Abschnitten zur Veranschaulichung der einzelnen Operationen anhand von Beispielen benötigt. Eingabedaten über die Extremitäten sind bei einer Benutzung von elektromagnetischen Sensoren und *Cyber*gloves gegeben durch:

1. Position und Ausrichtung definiter Körperpunkte, beispielsweise als 4x4 homogene Transformationsmatrizen
2. Stellung der einzelnen Gelenke der Finger und Hände als je ein nummerischer Wert

Dieses sind jeweils exemplarische Rohdaten*records* aus den über die Samplingraten der Sensorik diskretisierten Eingabewerten für die Gestenerkennung und -analyse. Eine Anforderung an das hier entwickelte System stellt die Forderung nach Unabhängigkeit von der eingesetzten Sensorik und den damit verbundenen Datenformaten dar. Letztere können, wie beschrieben, als nummerische Daten über Gelenk- oder Körperkonfigurationen vorliegen, aber je nach Verarbeitungsphase auch durch kombinatorisch oder hierarchisch angeordnete Berechnungsschritte ein grundsätzlich anderes Format besitzen. In einem solchen Fall sind vorverarbeitete Daten beispielsweise durch Attribute bzw. durch nummerisch angereicherte Attribute repräsentiert. Ein Erkennungs- und Analysesystem muss diesen verschiedenen Umständen Rechnung tragen. Es sollte unabhängig von der zur Verfügung stehenden Rate sein und muss sich adaptiv auf verschiedene, die Berechnungen initiierende Pulsdaten einstellen lassen und dabei keine Annahmen über das Format der zu verarbeitenden Daten machen.

Erkennung und Analyse in der VR-Loop

Grundsätzlich werden die Analyseergebnisse als Interaktionsgrundlage in einer VR-Anwendung dienen. Durch zwei unterschiedliche Interaktionsabläufe

(s. dazu Abschnitt 3.2) müssen zwei Arten der technischen Kopplung gestischer Analyse in das VR-System zur Verfügung gestellt werden. Dabei sind die in Abschnitt 2.3 erörterten Randbedingungen eines VR-Systems zu berücksichtigen. Es müssen zum einen Möglichkeiten loser Kopplung für event-basierte sprach-gestische Erkennungsaufgaben, zum anderen — besonders bei der gestischen Umsetzung von Interaktionen — Methoden einer engen Kopplung der *Bewegungs*analyse im Kontext einer Render-Loop gefunden werden. Letztere Bedingung erfordert sowohl eine homogene Einbettung der entwickelten Verfahren in Bezug auf die gewählte Implementationssprache, als auch die Möglichkeit zum Anschluss der Berechnungspropagation an externe Applikationslogiken und Abläufe. Hier muss eine geordnete Ausführung der Gestenerkennung durch eine Kopplung an den Ablauf einer speziellen Phase in der Render-Loop ermöglicht werden.

Weiterhin ist das Synchronisationsproblem zwischen der Render-Loop und zwischen den unterschiedlichen Datenquellen zu berücksichtigen. Diese sind i.A. sehr unterschiedlicher Natur. Aus technischer Sicht erfolgt das Rendering zu festgelegten Zeitpunkten, angestrebt wird eine feste Frequenz. Daher werden in der App-Phase verankerte Verfahren mit eben dieser Frequenz ausgeführt. Eine hier eingekoppelte Gestenerkennung basiert allerdings auf Sensordaten, welche durchaus verschiedene Samplingraten besitzen und somit einer unterschiedlichen Taktung unterliegen. Neben der Unabhängigkeit von diesen Datenraten bedingt eine genaue quantitative Auswertung der Gestik allerdings eine Gewährleistung einer garantierten Kontinuität in der Datenrate oder mindestens der Bereitstellung eines Distanzmaßes zwischen zwei Datenpaketen. Wird eine Propagation der Auswertung an den Ablauf der Render-Loop mit ihrem eigenen Synchronisationstakt gekoppelt, so ist eine Asynchronität zwischen beiden sehr wahrscheinlich. Hier müssen die eingesetzten Verfahren in jedem Fall Pufferungsmechanismen vorsehen.

4.2 Aktuatoren

4.2.1 Konzeptbasis

Soll im Rahmen multimodaler Interaktion gestische Eingabe für eine VR-Anwendung etabliert werden, so müssen zwei verschiedene Funktionen zur Verfügung gestellt werden. Erstens ist es erforderlich, eine Analyse der spatiotemporalen Bewegungsmerkmale der Benutzergestik zu gewährleisten, um so eine *gestische* Basis für eine Merkmalsdetektion und eine darauf aufsetzende multimodale Integration zur Verfügung zu stellen. Zweitens müssen Informationen über die gestische Artikulation des Benutzers echtzeitfähig und mit möglichst geringer Latenzzeit im Zusammenhang der gerade aktuellen Szenenkonfiguration bereits teilweise interpretiert werden (z.Bsp. bei der Deixisauswertung), da es einen direkten Bezug zwischen sprach-gestischen Äußerungen und der aktuellen spatialen Umgebung (im Sinne einer VR-Modellierung: der aktuellen Szenenkonfiguration) aus Sicht des Artikulierenden gibt (s. Abschnitte 2.2.2 und 2.4.1).

Darüber hinaus ist es vorgesehen, spezielle Bewegungsformen — kinemimische/mimetische Gesten — in der Anwendung zur vermittelten Steuerung, dem Vormachen von gewünschten Manipulationen zu verwenden. Dieses erfordert eine mit der Gestik schritthaltende Analyse und Erkennung, welche nicht auf einer bereits *abgeschlossenen* Klassifikationen operiert, sondern während des Verlaufs die Erkennung und die Interpretation der Geste ermöglicht.

Zur Bereitstellung dieser Funktionen erweisen sich bisherige Verfahren der Eingabeverarbeitung als nur eingeschränkt verwendbar. Grundlage dieser sind die Informationen über (geänderte) physikalische Konfigurationen eines externen Geräts — etwa des Drückens eines Schalters oder der Widerstandsänderung eines Potentiometers. Diese Daten werden als gepufferte atomare Signale — eventbasiert — in einem gesonderten Fetch-Zyklus einer Anwendung zur Verfügung gestellt. Der Kontext ihrer Auswertung wird allein durch den aktuellen Anwendungs*zustand* (applicationstate) definiert.

Kontext der Eingabeverarbeitung

Die Eingabeverarbeitung erfolgt in bisherigen Systemen[1] über sogenannte objektzentrierte Message-Loops (Nachrichtenschleifen). In diesen Endlos-

[1] Hier wird Bezug auf die Modellierung interner und externer Nachrichtenübermittlung in fensterbasierten WIMP-Systemen genommen, welche betriebssystemübergreifend eingesetzt werden.

schleifen[2] werden jegliche Eingaben in Form von speziellen Nachrichten an die Anwendungsobjekte zugestellt. Obwohl diese Form der Nachrichtenübermittlung an sich nicht darauf beschränkt ist, soll hier besonderes Augenmerk auf die Behandlung der Ein/Ausgabeoperationen gelenkt werden. In Bezug auf die Eingabe handelt es sich bei den Nachrichten zum Beispiel um den Status der Maustasten und die Mausposition oder eventuell gedrückte Tasten auf dem Keyboard. Hardwareseitig werden diese physikalischen Geräte mit einem definierten Sampletakt abgefragt, die analogen Werte daraufhin diskretisiert und anschließend zur Gewährleistung einer Synchronität und Datenkonsistenz in Pufferspeichern abgelegt. Diese Daten werden daraufhin in der Schleife ausgelesen und dem entsprechenden *aktiven* ausgabebezogenen Objekt, dem Fenster, dem Menü oder dem Icon, welches den *Fokus* besitzt, als Nachricht zugestellt. Die Eingabefokussteuerung wird i.A. durch einen endlichen Automaten modelliert, in welchem es zu einem bestimmten Zeitpunkt nur ein solches aktives Objekt gibt. So ist der Kontext für die Interpretation der übermittelten Daten nur vom aktuellen internen Zustand und damit vom aktiven Objekt abhängig. Dieser ändert sich in definierten Schritten (den applicationstates) und ist zwischen den Zustandsänderungen als statisch anzusehen.

Anforderungen und Spezifikation

Der direkt sensorisch zugängliche Kontext der sprach-gestischen Äußerungen, das Bezugs- und Referenzsystem, ist der den Benutzer umgebende Raum. In der VR ist dieses die virtuelle Szene mit den dort dargestellten Objekten und deren impliziten Referenzsystemen **relativ** zum Benutzerstandort. In immersiven Anwendungen sind die geometrischen Beziehungen zwischen Benutzer und Szene hochgradig dynamisch und variabel; eine Interpretation der Eingabedaten ist abhängig von der aktuellen räumlichen Konfiguration *Benutzerposition–Szenensicht–Szene*. Diese Informationen stehen erst im Kontext der Render-Loop der laufenden VR-Anwendung zur Verfügung. Darüber hinaus gibt es im Gegensatz zur geschilderten Verarbeitung von Maus- und Tastatureingaben in WIMP-basierten Systemen bei gestischen (und sprachlichen) Eingaben eine starke interpretative Komponente: Die Nachricht über das Vorliegen einer bestimmten Geste muss erst einmal aus den Gestendaten ermittelt werden. Dieses erfolgt einerseits teilweise unter der Auswertung externer — auf die aktuelle virtuelle Szene bezogener — Referenzen und Symmetrien. Andererseits handelt sich bei einer Geste nicht prinzipiell um ein atomares Ereignis, welches eventbasiert verarbeitet werden

[2] "Endlos" ist relativ zu der Laufzeitdauer der die Verarbeitung steuernden Anwendung.

kann[3], sondern um eine Körperhaltungsänderung, welche über ein Zeitintervall durch die genannten spatiotemporalen Merkmale (s. Abschnitt 4.1.2) bestimmt ist.

Weiterhin basieren diese Merkmale (s. Abschnitt 4.1.1) entweder auf bestimmten Körperpunkten und ggfs. dort impliziten Referenzrichtungen (s. Abbildung 4.1 auf S. 69) oder aber auf Formbeschreibungen (z.Bsp. der Hände). Diese Informationen stehen in den wenigsten Fällen direkt auf Grund der verwendeten Sensorik und ihren Befestigungspunkten zur Verfügung. Vielfach sind erst geometrische Transformationen oder Normierungen über Vergleiche mit Kalibrierungswerten notwendig, um aus den rohen Sensordaten Werte mit einem für die Gestenerkennung und die gestische Auswertung benötigten *semantischen* Bezug zu generieren, z.Bsp. die Ausrichtung der rechten Hand statt des Wertes eines 6DOF-Sensors auf dem rechten Arm oder die Fingerbeugung des Zeigefingers statt der rohen nummerischen Werte zweier Bimetallstreifen.

Um eine auf Regeln operierende Merkmals- und Gestenerkennung in VR-Anwendungen zu realisieren, müssen die genannten Unterschiede zu einer herkömmlichen eventbasierten Eingabeverarbeitung berücksichtigt werden. Ein besonderes Problem ergibt sich durch die Anforderung, verschiedene Datenquellen, i.A. durch Samplewerte eines physikalischen Gerätes gegeben, gemeinsam auszuwerten und zu fusionieren. Stimmen die Samplezeitpunkte der Geräte nicht überein, stellt sich die Frage nach geeigneten *Auswertezeitpunkten*. Wann wird im Kontext der geschlossenen Render-Loop die Erkennung und Analyse der Gestik ausgeführt, ohne dabei auf evtl. ungültige, da zum aktuellen Zeitpunkt nicht definierte[4] Sensordaten zurückzugreifen? Wie oft soll oder muss eine solche Gestenerkennungsfunktion ausgeführt werden? Wie geschieht ein Zugriff auf externe geometrische Referenzsysteme? Auf welchem Ausgabeformat wird das Erkennungs- und Analysesystem operieren? Und wie werden interaktive Elemente eingebunden?

[3]Dies steht im Gegensatz zu den verschiedenen möglichen Erkennungsevents (s. Abbildung 4.13)

[4]Die Sensorik arbeitet in diskreten Zeitschritten. Streng genommen sind die Sensorwerte nur für diese einzelnen Samplezeitpunkte definiert.

Zusammengefasst muss von rohen Sensor- und Eingabedaten abstrahiert werden und sowohl eine die Datenkonsistenz und Wertegültigkeit erhaltende Vorgehensweise für die zeitliche Synchronisation, als auch die Bereitstellung einer eindeutigen Bezugsbasis für die Gestenerkennung und -analyse ausgearbeitet werden. Die sensorisch erfassten Daten müssen

- normiert (gemeinsames Referenz- und Bezugssystem),
- synchronisiert (gemeinsame Sample- und Auswerterate),
- verrechnet, also beispielsweise
  - transformiert und/oder
  - verknüpft und/oder
  - klassifiziert und
- qualitativ annotiert werden.

4.2.2 Definition des Abstrakten Aktuators

Um die gestellten Anforderungen zu erfüllen, werden spezielle abstrakte Komponenten entwickelt. In Anlehnung an Begriffe aus der Regelungstechnik werden sogenannte *Abstrakte Aktuatoren* eingeführt. Als *Aktuator* wird dort die Gesamtheit von Entitäten bezeichnet, welche für die Impulsweitergabe an einen *Endeffektor* verantwortlich ist. Ein solcher Endeffektor in dem hier gebräuchlichen Sinn repräsentiert den Ausschnitt der menschlichen Körperbewegung, welcher für die Identifikation bestimmter Gesten ausreichend ist. Abstrakte Aktuatoren kapseln alle bis dahin anfallenden sensorischen Vorberechnungen und bieten eine einheitliche Schnittstelle für weitere Auswertemodule.

Definition 8 (Abstrakter Aktuator) *Abstrakte Aktuatoren sind von Sensorik entkoppelte Container von qualitativ annotierten und für gestische Informationsvermittlung signifikanten Körpermerkmalen in einer einheitlichen Bezugsbasis. Ein Abstrakter Aktuator (kurz: ${}_2A$)ist ein 4-Tupel*

$$ {}_2A = (\mathcal{K}_\mathcal{E}, \sigma, \mathcal{N}, \beta) \tag{4.1} $$

Dabei sei

- $\mathcal{K}_\mathcal{E} = \{e_1, \ldots, e_i\}$ eine Menge von *Eingabekanälen*

  $\mathcal{K}_\mathcal{E} \subset (\mathbb{R} \mapsto \mathcal{P}(\mathbb{R} \times \mathcal{E}_j)) \times (\mathbb{R} \mapsto \mathcal{E}_j)$ mit
  $e_j = (S_j(t), \vec{\kappa}_j(t)), j \in \{1, \ldots, i\}$, ein Tupel aus

  $S_j : \mathbb{R} \mapsto \mathcal{P}(\mathbb{R} \times \mathcal{E}_j)$
  $S_j(t) = \{(t_x^j, \vec{\delta}(t_x^j)) \mid t_x^j \in T_j(t)\}$, einer Menge von Samplewerten des Kanals e_j zum Zeitpunkt t mit

  $T_j : \mathbb{R} \mapsto \mathcal{P}(\mathbb{R})$
  $T_j(t) = \{t_k^j < t_{k+1}^j < \ldots < t_0^j \leq t, t_i^j \in \mathbb{N}_0, k \in \mathbb{Z}_0^-\}$, einer zeitabhängig wachsenden endlichen Menge von diskreten, disjunkten und geordneten Zeitpunkten und

  $\vec{\delta}_j : T_j \mapsto \mathcal{E}_j$, einer diskreten Wertzuordnungsfunktion $\vec{\delta}_j(t_x^j)$, welche den zum Zeitpunkt t_x^j gehörigen Samplewert aus dem Kanaleingaberaum $\mathcal{E}_j$ liefert

  und

  $\vec{\kappa}_j : \mathbb{R} \mapsto \mathcal{E}_j$, einer kontinuierlichen Zugriffsfunktion mit

  $$ \vec{\kappa}_j(t) = \begin{cases} \vec{\iota}_j(t), \text{ einer geeigneten Interpolationsfunktion} & \\ \text{wenn } t_k^j \leq t \leq t_0^j & \text{oder} \\ \vec{\chi}_j(t), \text{ einer Extrapolationsfunktion} & \text{sonst.} \end{cases} $$

- ...

weiter $\longrightarrow$

- σ eine *Synchronisationsfunktion* zwischen den Eingabekanälen

$$\sigma : \begin{cases} 1. \ (\mathrm{R} \times \mathcal{P}(\mathbb{R} \mapsto \mathcal{P}(\mathbb{R}))) \mapsto \mathcal{P}(\mathbb{R}) \text{ ungekoppelt variabel} & \text{oder} \\ 2. \ (\mathrm{R} \times \mathcal{P}(\mathbb{R} \mapsto \mathcal{P}(\mathbb{R})) \times \mathbb{R}) \mapsto \mathcal{P}(\mathbb{R}) \text{ ungekoppelt fest} & \text{oder} \\ 3. \ (\mathrm{R} \times \mathcal{P}(\mathbb{R} \mapsto \mathcal{P}(\mathbb{R})) \times (\mathbb{R} \mapsto \mathcal{P}(\mathbb{R}))) \mapsto \mathcal{P}(\mathbb{R}) \text{ gekoppelt} & \text{sonst.} \end{cases}$$

  Sei
  $t^{min}(t) = max\{t_k^j \mid j = 1 \dots i, t_k^j \in T_j(t)\}$
  der früheste Zeitpunkt, in welchem für *jeden* Kanal Werte vorliegen,
  $t^{max}(t) = min\{t_0^j \mid j = 1 \dots i, t_0^j \in T_j(t)\}$
  der späteste Zeitpunkt, in welchem für *jeden* Kanal Werte vorliegen.
  Dann sind die möglichen σ:

  $$\sigma_1(t, \{T_1(t), \dots, T_i(t)\}) = \{t' \mid t^{min}(t) \leq t' \leq t^{max}(t)\}$$

  $$\sigma_2(t, \{T_1(t), \dots, T_i(t)\}, i) = \{t'_x \mid t^{min}(t) \leq t'_x \leq t^{max}(t) \wedge \forall x > k : t'_x - t'_{x-1} = i, \text{ mit } i \in \mathbb{N}\}$$

  $$\sigma_3(t, \{T_1(t), \dots, T_i(t)\}, T_r(t)) = \{t'_x \mid t^{min}(t) \leq t'_x \leq t^{max}(t) \wedge t'_x \in T_r(t)\}$$

  Dabei sei i eine Intervalllänge und $T_r, r \in \{1, \dots, i\}$ die Menge diskreter Samplezeitpunkte eines Referenzkanals.

- $\mathcal{N} = \{\vec{\nu}_1, \dots, \vec{\nu}_i\}$ eine Menge von *Normierungsfunktionen*
  $\vec{\nu}_j : \mathcal{E}_j \mapsto \mathcal{E}_j$, welche Vektoren aus gleichen $\mathcal{E}_j$ in eine gemeinsame Basis bringen und für diese ein eindeutiges Abstandsmaß definieren.

- $\beta : \mathbb{R} \mapsto (\mathcal{A}_1 \times \dots \times \mathcal{A}_o)$
  $(\vec{a}_1(t), \dots, \vec{a}_o(t)) = \beta(t)$
  eine zeitabhängige Berechnungsfunktion für $o \in \mathbb{N}$ Ausgabewerte
  $\vec{a}_l : \mathbb{R} \mapsto \mathcal{A}_l$ der *Attributsequenzen*

  $A_l : \mathbb{R} \mapsto (\mathcal{P}(ASCII) \times \mathcal{A}_l)$
  $A_l(t) = (\texttt{<Bezeichner>}_l, \vec{a}_l(t'))$, mit $l \in \{1, \dots, o\}$, $t' \in \sigma, t' \leq t$,
  Tupeln aus Bezeichner und einer diskreten Wertzugriffsfunktion.

Die zeitabhängigen Samplewertmengen $S_j, j = 1 \ldots i$ der entsprechenden Kanäle e_j sind i.A. durch die Hardware der eingesetzten Sensorik gegeben. Abstrakte Aktuatoren fassen o.B.d.A. unterschiedliche Kanäle zusammen und liefern mit Hilfe der Synchronisationsfunktion σ mögliche Auswertezeitpunkte, in denen eine Interpolation ι_j über die sonst in der Regel von der Hardware gegebenen asynchronen Samplezeitpunkte T_j ermöglicht wird. σ liefert als Ergebnis ein Zeitintervall, in welchem zu jedem Zeitpunkt die somit kontinuierlich zugängliche Berechnungsfunktion β die verschiedenen Kanalinformationen verarbeiten und fusionieren kann. Diese spezielle Funktionalität ist abhängig vom ${}_2A$-Einsatzzweck und wird in noch folgenden Abschnitten für konkrete Beispiele erläutert.

Eine Extrapolationsfunktion χ wird spezifiziert, um etwaige Latenzzeiten zwischen Samplezeitpunkt und der Weiterverarbeitung in echtzeitkritischen Modulen zu verkürzen. Ein Beispiel dafür ist die *Vorausberechnung* der Position und Ausrichtung des Kopfes eines Benutzers, um so eine möglichst verzögerungsfreie Projektionsberechnung für die benutzerabhängige Bildgenerierung zu ermöglichen. Da eine solche Vorgehensweise sich nur auf wenige Daten über die vollzogene Bewegung und einige Heuristiken über den wahrscheinlichen Verlauf stützen kann, ist sie auf Grund der Fehleranfälligkeit nicht als Datenlieferant für eine Gestenerkennung und -analyse geeignet. Besonders bei der Gestenanalyse kommt es auf möglichst präzise Informationen über die Benutzerbewegungen an, damit die eingesetzten Verfahren — z.Bsp. einer auf geometrischen Bezügen arbeitenden Trajektorienbewertung — nicht unbrauchbar werden.

Die Normierungsfunktionen $\mathcal{N}$ dienen dazu, Ausgaben in einheitlichen Wertebereichen und Bezugssystemen zu generieren. Relative Beschreibungen von Körpermerkmalen, welche durch eine Maximal- und eine Minimalausprägung gegeben sind — das sind beispielsweise Streckungsattribute über Gelenkwinkelwerte — werden einheitlich auf $[0, 1]$ Intervalle abgebildet. Rein geometrische Informationen aus dem $\mathbb{R}^3$ dagegen können nur in einer gemeinsamen Basis sinnvoll zueinander in Bezug gesetzt werden. Diese Abbildungsfunktionen aus den Sensorwertebereichen werden durch die kanalspezifischen ν_j umgesetzt.

Die Synchronisationsfunktion σ liefert zu einem beliebigen Zeitpunkt t eine Menge von möglichen Zeitpunkten t', zu denen eine auf Interpolationsdaten operierende Ausgabe möglich ist. Abbildung 4.2 verdeutlicht an Hand

des speziellen Falls von *festen* Kanalraten, also äquidistanten Samplezeitpunkten, die Problematik bei dem Versuch, eine unabhängige Auswerterate t' allein auf diskreten Zeitpunkten zu stützen.

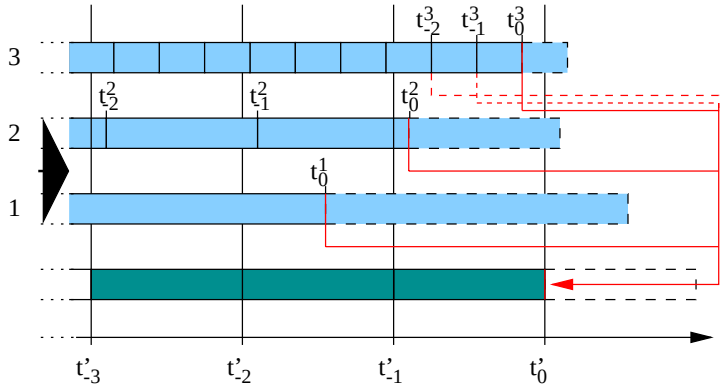

Abbildung 4.2: Diskrete Kanalsynchronisation mit drei verschiedenen Möglichkeiten von Samplezeitpunkten (im speziellen Fall *feste* Taktraten) zu einem Auswerteintervall.

Das Beispiel skizziert die drei Möglichkeiten des Bezugs von Samplezeitpunkten der Eingangskanäle zu einer Ausgabe- und Auswertezeit. Für nicht mittels Hardware synchronisierte Kanäle sind zu einem beliebigen Zeitpunkt t'_0 prinzipiell keine aktuellen Samplewerte zu erwarten. Je grösser die Latenzzeit ist, also der Abstand zwischen aktuellem Zeitpunkt und dem letzten gültigen Samplewert, umso fehlerbehafteter ist eine etwaige Extrapolation. Des Weiteren wird je nach Ratenverhältnis entweder ein Super- (Kanal 1) oder ein Subsampling (Kanal 3) nötig sein. Im letzten Fall muss eine Subsamplefunktion ϕ die Menge der entsprechenden Eingabevektoren im Ausgabeintervall $]t'_{-1}, t'_0[$ auf einen Eingabevektor abbilden. Mögliche Alternativen (bezogen auf Kanal 1 in Abbildung 4.2), die Samplefunktion ϕ zu definieren, wären o.B.d.A. :

$$\phi\{\vec{\delta}_3(t^3_y) \mid t'_{-1} < t^3_y \leq t'_0\} = \begin{cases} \vec{\delta}_3(t^3_0) & \text{oder} \\ \vec{\delta}_3(t^3_x), \nexists\, t^3_z : \; t'_{-1} < t^3_z < t^3_x & \text{oder} \\ \varnothing\{\vec{\delta}_3(t^3_y) \mid t'_{-1} < t^3_y \leq t'_0\} & \text{oder} \\ \ldots \end{cases} \tag{4.2}$$

Jede der angegebenen Möglichkeiten für ϕ hat dabei ihre eigene Schwäche. Entweder werden unter Umständen wichtige Samplewerte unterschla-

gen, oder es wird die Latenzzeit unnötig erhöht oder die Daten unterliegen einer unerwünschten Glättung (sofern ϕ über Durchschnittsbildung berechnet wird; s. Fall $\varnothing$).

Durch σ und die entsprechenden κ_j werden dagegen a) gültige Wertebereiche für Interpolationen und b) eine darauf basierende kontinuierliche Auswertung ermöglicht und c) die Problematik eines Subsampling vermieden. Eine zu niedrige Samplerate (s. Kanal 1) wird dagegen grundsätzlich problematisch sein, da der letzte Wert als obere Schranke für $T'(t)$ dient und damit immer maßgeblich für die Latenzzeit zeichnet.

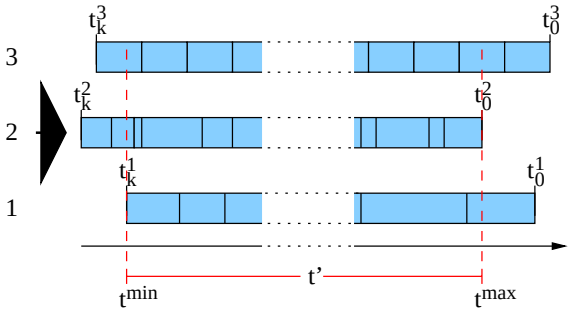

Abbildung 4.3: Ungekoppelte variable Synchronisation über eine kontinuierliche Menge von Zeitpunkten t' im Intervall gegeben durch den möglichen Maximalwert t_0 im Kanal 2 und den Minimalwert t_k im Kanal 1.

Die angegebenen drei möglichen Umsetzungen für σ schränken die Menge $T'(t)$ in ansteigender Reihenfolge weiter ein und sind dem gegebenen Einsatzzweck anzupassen. Wird der Abstrakte Aktuator *ungekoppelt variabel* betrieben (σ_1), so repräsentiert $T'(t)$ ein kontinuierliches Zeitintervall, welches nur durch die obere und untere Schranke begrenzt ist (s. Abbildung 4.3). Eine darauf fußende Anwendung muss Sorge dafür tragen, die Aktuatorfunktionen zu bestimmten Zeitpunkten auszuführen und wird dabei anstreben, die jeweils obere Schranke des Intervalls als Auswertezeitpunkt zu wählen, wenn aktuelle Informationen generiert werden sollen.

Eine feste, von den Kanalsamplezeitpunkten unabhängige Auswertung arbeitet in einem eigenen Takt (σ_2). Für diesen Fall ist weiterhin ein Zeitpunkt für die jeweilige Lage eines Referenzintervalls festzulegen, im gegebenen Beispiel in Abbildung 4.4 fällt t^{max} auf das Ende des aktuellen Intervalls, so ergibt sich eine minimale Latenzzeit. Dieses kann allerdings im Gegensatz zum gegeben Beispiel nur in Bezug auf einen festen Referenztakt garantiert werden, da ansonsten keine Annahmen über die zeitabhängigen t^{min} und t^{max} gemacht werden können.

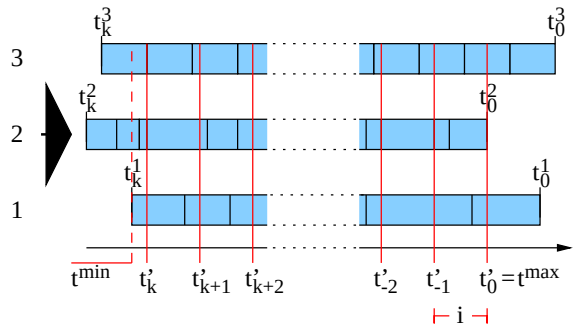

Abbildung 4.4: Ungekoppelt feste Synchronisation in einem Intervall der Länge i im Beispiel beginnend an der oberen Schranke des möglichen Gesamtintervalls.

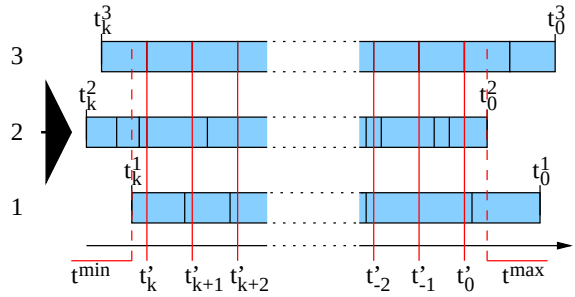

Abbildung 4.5: Gekoppelt feste Synchronisation an einen gegebenen Referenzkanal (Kanal 3) mit etwaiger Latenzzeit in Bezug auf t^{max}.

Werden mehrere Abstrakte Aktuatoren betrachtet, so unterscheiden sie sich in erster Linie in der Festlegung der Funktionen σ. Für unterschiedliche ${}_2A$ sollten die σ i.d.R. gleiche Eigenschaften implementieren, um in einer Anwendung eine einheitliche Behandlung in Bezug auf die Synchronisation der Sensordatenverarbeitung zu gewährleisten. Die ν_j arbeiten kanalzentriert und müssen für gleiche $\mathcal{E}_j$ in gemeinsame Basen transformieren, um übergreifende Bewertungen zu ermöglichen. Die Spezialisierung der Berechnungsfunktion β unterscheidet dagegen die Aktuatoren gemäß der konkreten Elemente aus den $\mathcal{A}_1 \ldots \mathcal{A}_o$. Beispiele dafür werden in den noch folgenden Abschnitten vorgestellt, zuvor werden Methoden ausgearbeitet, um die Abstrakten Aktuatoren in einer VR-Anwendung zu verankern.

4.2.3 Aktuatorknoten

Um die Funktion der Abstrakten Aktuatoren im geschlossenen Ablauf einer Render-Loop und dem geometrischen Bezugssystem einer virtuellen Szene zur Verfügung zu stellen, werden **Aktuatorknoten** als eine neue Knotenklasse in den Applikations- und Szenengraph — kurz AS-Graph — eingeführt (s. Abbildung 4.6 auf S. 91). Diese kapseln die Aktuatorfunktionalität, reichern sie mit den für Szenengraphkomponenten benötigten Eigenschaften an und dienen als Vermittlerobjekte zwischen Sensorik und Gestenerkennung.

Definition 9 (Aktuatorknoten) *Aktuatorknoten (Act-nodes) sind um Aktuatorfunktionalität angereicherte Szenengraphkomponenten mit Gruppierungs- und Transformationseigenschaften. Ein Aktuatorknoten ist ein 6-Tupel*

$$AK = ({}_2A, p, C, M, T_{eval}(t), FM(t)) \tag{4.3}$$

Dabei sei

- ${}_2A$ ein Abstrakter Aktuator mit der damit verbundenen Funktionalität (n. Definition 8).
- p ein Vaterknoten (parent) im Szenengraphen.
- $C = \{c_1, \ldots, c_n, n \in \mathbb{N}\}$, eine Menge von Kinderknoten (children).
- $M \in \mathcal{M}(4, 4, \mathbb{R})$ eine homogene Transformationsmatrix (4x4 Matrix über dem Körper $\mathbb{R}$).
- $T_{eval}(t) = \{t_k^e < t_{k+1}^e < \ldots < t_0^e \leq t, t_i^e \in \mathbb{N}_0, k \in \mathbb{Z}_0^-\}$ eine zeitabhängig wachsende endliche Menge von diskreten, disjunkten und geordneten Traversierungszeitpunkten im Ablauf der Render-Loop.
- $FM(t)$ eine *Menge* von o Ausschnitten aus den Attributsequenzen des Abstrakten Aktuators im letzten Traversierungsintervall relativ zum Zeitpunkt t:

  $$FM(t) = \{F_l(t'_x) \mid l \in \{1, \ldots o\}, t'_x \in \sigma(t_{y-1}^e, \ldots) \cap \sigma(t_y^e, \ldots), \\ t_y^e \in T_{eval}(t) \wedge \nexists\, t_z^e \in T_{eval}(t) : t_y^e < t_z^e \leq t\}$$

Aktuatorknoten bilden den Link zwischen der Definition Abstrakter Aktuatoren und den darin spezifizierten Funktionen mit der Struktur und Ablaufsteuerung einer VR-basierten Anwendung durch die zwei folgenden Zusammenhänge:

a) Die $\nu_j(M \cdot_p T^{root}, \dots)$ Normierungsfunktionen ν_j sind für $\mathbb{R}^3$ basierte Sensordaten[5] abhängig vom Referenzsystem (der Basis) der virtuellen Szene gegeben durch die Einordnung in den Szenengraphen (s. Abbildung 4.6) und die damit bestehende Gesamttransformation zum Aktuatorknoten $M \cdot T_p$.

b) $\sigma(t_x^e, \dots), t_x^e \in T_{eval}(t))$ Die möglichen ${}_2A$-Ausgabezeitpunkte sind abhängig von $T_{eval}(t)$. Das entspricht einer Kopplung der Auswertung an die Graphtraversierung im Zuge der Renderphasen. Diese Bedingung gewährleistet konsistente Szeneninformationen (s. **a)**) für jedes neu zu berechnende Bild, also einer Ausführung innerhalb vollständiger *Frameboundaries*.

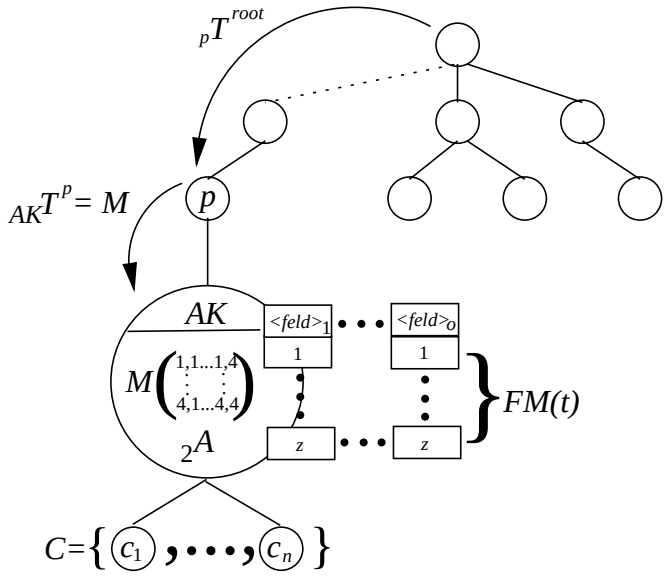

Abbildung 4.6: Aktuatorknoten *AK* mit *parent p* und *children C* eingebettet im Szenengraphen. Die zeitabhängige Anzahl von Ausgabefeldmengen <feld>$_1 \dots$ <feld>$_s$ (als Ausschnitt der Sequenzen) ist angedeutet. ${}_xT^y$ transformiert Punkte zur Basis x in das Referenzsystem y.

[5] Ein Grossteil der benötigten sensorischen Werte liefert Daten über die Bewegungen im $\mathbb{R}^3$.

Die Betrachtung der einzelnen Kanalsamplezeitpunkte $T_1(t), \ldots, T_i(t)$ und der darauf basierenden Synchronisationsfunktion $\sigma(t)$ unter Berücksichtigung der Traversierung innerhalb der Render-Loop und der damit verbundenen Auswertung — vom Einlesen reiner Rohdaten der Sensorhardware bis hin zur Generierung symbolisch annotierter Feldwerte — verdeutlicht die inhärenten Schwierigkeiten die unterschiedlichen Steuerungs-, Kontroll- und Sampleabläufe in einheitlicher Weise zu vereinen. Die Aktuatorenkonzepte stellen in diesem Zusammenhang benötigte Lösungsmöglichkeiten zur Verfügung.

Das Synchronisationsproblem

Durch die Modellierung unter Einbeziehung der σ und der $T_x(t)$ wird das Synchronisationsproblem zwischen geschlossener Render-Loop und der Einbindung externer Datenquellen, welche in eigenen Zyklen abgefragt werden, durch die formale Betrachtung explizit gemacht. Die benötigten Funktionen arbeiten prinzipiell zeitabhängig, da — wie sich auch in der Realität gezeigt hat — keine allgemein gültigen Angaben über die Samplezeitpunkte (z.Bsp. die Annahme einer *festen Taktrate)* von eingesetzter Sensorhardware vorauszusetzen sind. So erfüllen schon die unterschiedlichen kommerziellen Sensortypen diese notwendig Voraussetzungen in keiner Weise. Die gleiche Problematik kann sich aber ebenfalls prinzipiell ergeben, z.Bsp. bei in der Entwicklung stehenden optischen Verfahren[6], welche je nach Bildinhalt — also kontextabhängig — unterschiedliche Verarbeitungszeiten benötigen. In solchen Fällen kann nur die hier konzipierte aktuelle Betrachtung von einem zum nächsten Auswertezeitpunkt erfolgen. Daneben liefert das Verhältnis von t_0' zu t die zu erwartende Gesamtlatenzzeit über die verschiedenen Sensorquellen hinweg. Da es sich bei VR-Systemen um echtzeitbasierte Systeme handelt, sind die Latenzzeiten für eine dabei notwendige Erwartbarkeitsbetrachtung notwendig. Hier wird nicht angestrebt möglichst schnelle Antwortzeiten zu erhalten, sondern es sollen Abschätzungen ermöglicht werden, innerhalb welcher Zeitspanne die Antwortzeit liegt. In VR-Systemen wird auf diese Weise eine Einschätzung der minimalen Framerate ermöglicht. Zum anderen erlauben die vorgeschlagenen Varianten von σ eine geeignete Parametrisierung und damit eine adäquate Anpassung an die jeweilig geltenden Anforderungen sowohl der Gestenerkennung und -analyse als auch der Render-Loop. σ spezifiziert entweder wohldefinierte Bereiche oder konkrete Zeitpunkte, in welchen Interpolationsfunktionen eine kontinuierliche Ausführung der Auswertefunktion β gestatten.

[6] Die hier entwickelten Konzepte berücksichtigen ebenfalls die Einbindung von im Zuge eines Kooperationsprojekts (s. [77] und [78]) entwickelter optischer Sensorik.

Eine weitere Variable in diesem Synchronisationsprozess kommt durch die Einbindung in die Graphtraversierung zustande. Zwei Grenzwerte wurden in diesem Zusammenhang bereits genannt:

- Minimale Framerate für die Bildgenerierung = 20Hz
- Minimale Samplerate für die Gestenerkennung(-analyse) = 20Hz

Bezeichne $T_f(t)$ die Zeitpunkte der Framegenerierung innerhalb der VR-Anwendung, also die Zeitpunkte der Render-Loop-Durchläufe. Gilt also

$$\forall\ t_x^f \in T_f(t) \text{ mit } t_k^f < t_x^f < t_0^f : t_x^f - t_{x-1}^f = t_{x+1}^f - t_x^f, \tag{4.4}$$

so liegt eine feste Framerate[7] von $r_{frame} = \frac{1}{t_x^f - t_{x-1}^f}$Hz vor. Stellt man eine ähnliche Betrachtung für Samplezeitpunkte der Sensorhardware an, so ergibt sich auch in diesem Fall der Wunsch nach einem festen Sampletakt, also in Form einer Minimalforderung:

$$r_{frame} \gg 20 \tag{4.5}$$

$$r_{e_x} \gg 20 \text{ , mit } r_k \in r_{e_1}, \ldots r_{e_i} \tag{4.6}$$

Wie beschrieben ist die zweite Bedingung durch die Einführung einer Interpolationsfunktion insofern erfüllt, als durch diese beliebige Zeitpunkte $T'(t)$ über dem möglichen Auswerteintervall gewählt werden können (σ_1), bzw. gleich eine feste Taktrate eingestellt wird (σ_2 oder σ_3). Dadurch kann natürlich keine absolute Kontinuität in den Sensordaten garantiert werden. Datendrops oder punktuell ansteigende Latenzen, wie sie beispielsweise durch die Anbindung[8] der Sensorhardware erfolgen können, werden sich nur in Extremfällen negativ auf nachgeschaltete Module auswirken.

Durch die entwickelte Einbindung der Aktuatorknoten in den AS-Graphen wird weiterhin eine **lose kontinuierliche Kopplung** der Evaluationsrate mit der App-Phase unter Erhalt benötigter Szenenintegrität etabliert. Wird während der App-Phase der Aktuatorknoten traversiert, werden dessen anstehende (gegeben durch die Zeitpunkte gemäß σ) Berechnungsmethoden ausgeführt. Man kann sagen, die geschlossene Render-Loop entspricht hier in Bezug auf die gestische Eingabeverarbeitung der Message-Loop bei der

[7] Eine angestrebte Voraussetzung für VR-Anwendungen.

[8] Etwa eine nicht Echtzeit-basierte Abfrage serieller Schnittstellen oder durch eine Verwendung von unzuverlässigen Netzwerkprotokollen wie UDP (User Datagram Protocol).

Eingabebehandlung in den WIMP-basierten Systemen. Während der Graphtraversierung in der App-Phase werden die Sensordaten synchronisiert, normiert, ggfs. weiter verrechnet und weitergegeben.

Diese lose Kopplung verhindert eine Einflussnahme der Bildgenerierung oder der Simulationsdauer auf die Eingabeverarbeitung. So werden zwar Standardwerte für die Framerate r_{frame} im high-end VR Bereich um 48-60Hz angestrebt (für Desktop-basierte VR teilweise niedriger), können aber durch die Abhängigkeit zwischen der Renderdauer und den zu generierenden Bildinhalten sowie durch die schwer abschätzbaren zusätzlichen Aufgaben in der App-Phase (z.Bsp. durch die Simulationsdauer oder die Eingabeverarbeitung selbst) nicht garantiert werden. In bisherigen VR-Anwendungen konnte dieses Problem vernachlässigt werden. Dort werden Sensoreingaben benötigt, um entweder die benutzerspezifische Projektion zu berechnen oder um eine graphische Repräsentation eines Eingabegerätes — etwa einen virtuellen Stylusstrahl — oder ein manipuliertes Objekt auszurichten und zu positionieren. Da dieses natürlich nur für die gezeichneten Bilder geschieht ($T_f(t)$ ist hier defacto $T'(t)$), führt ein Einbruch in der Framerate nicht zu Missinterpretationen einer Eingabe. In solchen Anwendungen wird in der Regel eine **diskontinuierliche** Sensordateneingabe **entkoppelt** von der Render-Loop betrieben. Parallele Eingabeloops lesen in der von der Sensorhardware vorgegebenen Rate die neuen Sensorwerte aus und legen diese in einen für alle Verarbeitungsloops zugänglichen Speicherbereich. Werden diese Daten für das Rendering benötigt, so kann in diesem Speicher zu jedem Zeitpunkt das aktuelle Datum ausgelesen werden. Fallen pro Renderdurchgang mehr als ein neuer Sensorwert an, so werden diese überschüssigen Daten überschrieben und damit verworfen. Diese Art der asynchronen Verarbeitung resultiert daher in Brüchen im Eingabestrom und einer Abhängigkeit von der Framerate. Eine solche Kopplung ist im Kontext einer Gestenerkennung und -analyse absolut zu vermeiden.

Diesem Umstand wird mit den beschriebenen Konzepten Rechnung getragen. Eine variable Framerate mit etwaigen Einbrüchen unter den Minimalwert führt zu keinen Ausfällen in der Eingabedatenverarbeitung, sondern kann diese nur verzögern. Die Datenkonsistenz bleibt vollständig erhalten.

Begriffsvereinbarung: Für die folgenden Ausführungen werden die Begriffe Aktuator und Aktuatorknoten gleichgesetzt und damit die Knotenentitäten bezeichnet. Wenn speziell auf die Aktuatoren nach Definition 8 (s. S. 84) Bezug genommen wird, wird speziell der Begriff des *abstrakten* Aktuators Verwendung finden.

4.2.4 Aktuator- und Sequenzfeldtypen

Die explizite Kapselung der Aktuatorfunktionalität in Knotenentitäten sowie die Exportierung der Berechnungsergebnisse in entsprechenden Sequenzausschnitten ermöglicht a) eine Portabilität und Wiederverwendung durch die graphzentrierte Modellierung der Eingabeverarbeitung und b) eine standardisierte Schnittstelle für weiterverarbeitende Module respektive. Diese Vorgehensweise stellt einen Vorschlag dar, die bereits erörterten Standardisierungsbestrebungen für VR-basierte Modellierung (s. VRML97 [15], OSG [80] und Java3D) um Komponenten einer gestischen Eingabeauswertung zu erweitern.

Aktuatoren unterscheiden sich in der Art und dem Format der zur Verfügung gestellten Ausgabevektoren. Viele Aktuatoren liefern gleichzeitig mehrere unterschiedliche Sequenzfeldwerte, da diese beispielsweise auf den Rohdaten eines einzigen Kanals basieren und Mehrfachberechnungen in getrennten Aktuatoreninstanzen vom Gesichtspunkt der Effizienz zu vermeiden sind. Es folgen als Beispiele Beschreibungen von für die Gestenerkennung und -auswertung benötigten Aktuatoren und dabei Verwendung findende Sequenzfeldtypen. Auf die hier beschriebenen Aktuatoren wird darüber hinaus bei der Konzeption der Merkmalserkenner in den noch folgenden Abschnitten Bezug genommen.

Ein Handformaktuator

Sei $e_1 = (S_1(t), \vec{\kappa}_1(t))$ der Eingabekanal eines in der VR gebräuchlichen Datenhandschuhs[9] mit 18 Bimetallsensoren und einem Sensorwertebereich von $0 \ldots 255$ mit:

$$\begin{aligned} \vec{\kappa}_1 &: \mathbb{R} \mapsto \mathbb{E}^{18} \\ \mathbb{E}^{18} &= \{(v_1, \ldots, v_{18}) \mid v_x \in \mathbb{N}_0, 0 \leq v_x \leq 255\} \end{aligned} \tag{4.7}$$

Als Normierungsfunktion dient die lineare Abbildung in den gewünschten Wertebereich $0 \ldots 1$ durch:

$$\begin{aligned} \nu_1 &: \mathbb{E}^{18} \mapsto \mathbb{E}^{18} \quad \text{mit} \\ \mathbb{E}^{18} &= \{(v_1, \ldots, v_{18}) \mid v_x \in \mathbb{R}, 0 \leq v_x \leq 1\} \end{aligned} \tag{4.8}$$

$$\nu_1(\vec{\kappa}_1(t)) = \vec{\kappa}_1(t)/255 \tag{4.9}$$

[9] Im konkreten Prototypszenario ein Virtual Technologies *CyberGlove*.

Durch den Einsatz eines einzigen Kanals dieses Typs liegen bauartbedingt äquidistante Samplezeitpunkte, also eine feste Taktrate mit rund 100Hz vor. So wird hier als Synchronisationsfunktion σ_3 mit dem einzigen Kanal als Referenzkanal angewendet:

$$\sigma_3 : (\mathbb{R} \times \mathcal{P}(\mathbb{R} \mapsto \mathcal{P}(\mathbb{R})) \times (\mathbb{R} \mapsto \mathcal{P}(\mathbb{R}))) \mapsto \mathcal{P}(\mathbb{R}) \tag{4.10}$$

$$\sigma_3(t, \{T_1(t)\}, T_1(t)) = \{t'_x \mid t^{min}(t) \leq t'_x \leq t^{max}(t) \wedge t'_x \in T_r(t)\} \tag{4.11}$$

Bei dieser Art der Sensorik wird jeweils ein Bimetallsensor für die Beugungsmessung des Fingergrundgelenks und des Fingermittelgelenks eingesetzt. Laut Feldbezeichnung soll das Attribut *Fingerstreckung* ermittelt werden. Bei näherer Betrachtung tragen die Beugungswerte der beiden Gelenke nicht in gleichem Maße zu der Beurteilung eines gestreckten Fingers bei, sondern es erscheint besonders das Fingermittelgelenk als ausschlaggebend. Dieser Umstand wird in den Teilberechnungsfunktionen durch die Einführung eines Gewichtungsfaktors berücksichtigt. Der Aktuator führt analoge Berechnungen[10] für jeden Finger der Hand durch, da diese Daten gemeinsam durch einen einzigen Kanal übertragen werden. Für den Handformaktuator wird die spezifische Funktionalität β folgendermassen definiert:

$$\beta : \mathbb{R} \mapsto (\mathcal{A}_1 \times \cdots \times \mathcal{A}_5) \tag{4.12}$$

$$\begin{aligned}\beta(t) = (&\beta_t^{'}(\vec{\nu}_1(\vec{\kappa}_1(t))[a_t], \vec{\nu}_1(\vec{\kappa}_1(t))[b_t]),\\ &\beta_i(\vec{\nu}_1(\vec{\kappa}_1(t))[a_i], \vec{\nu}_1(\vec{\kappa}_1(t))[b_i]),\\ &\beta_m(\vec{\nu}_1(\vec{\kappa}_1(t))[a_m], \vec{\nu}_1(\vec{\kappa}_1(t))[b_m]),\\ &\beta_r(\vec{\nu}_1(\vec{\kappa}_1(t))[a_r], \vec{\nu}_1(\vec{\kappa}_1(t))[b_r]),\\ &\beta_p(\vec{\nu}_1(\vec{\kappa}_1(t))[a_p], \vec{\nu}_1(\vec{\kappa}_1(t))[b_p]))\end{aligned} \tag{4.13}$$

Dabei seien $x \in t, i, m, r, p$ Indizes, welche respektive für die englischen Bezeichnungen *thumb, index, middle, ring, pinkie* der einzelnen Finger vom Daumen an stehen und es wird zur Vereinfachung $\vec{v}_1(t) = \vec{\nu}_1(\vec{\kappa}_1(t))$ gesetzt. Dann werden die Teil-*Klassifikationsfunktionen* β_x für die relative Bewertung des Streckungsattributs folgendermaßen definiert:

[10] Die Berechnung $\beta_t^{'}$ für den Daumen wird leicht abgewandelt, da hier für die Streckung des Fingers das Fingergrundgelenk im Gegensatz zum Endgelenk unwichtig ist.

$$\beta_x(\vec{v}_1(t)[a_x], \vec{v}_1(t)[b_x]) = w_{a_x} * \left| \left(1 - \frac{\mid poscomp(a_x) - \vec{v}_1(t)[a_x] \mid}{\mid poscomp(a_x) - negcomp(a_x) \mid}\right) \right| + w_{b_x} * \left| \left(1 - \frac{\mid poscomp(b_x) - \vec{v}_1(t)[b_x] \mid}{\mid poscomp(b_x) - negcomp(b_x) \mid}\right) \right| \tag{4.14}$$

Dabei seien

- $a_x, b_x \in \{1, \ldots, 18\}$ Vektorindizes der den betrachteten Gelenken entsprechenden Sensoren,
- $(w_{a_x}, w_{b_x}) \in \{(w_a, w_b) \mid \forall\ w_a, w_b \in \mathbb{R}_0^+ : w_a + w_b = 1\}$, relative Gewichtungen der Gelenke für das entsprechende Attribut,
- $poscomp(c)$ ein positiver Vergleichswert und $negcomp(c)$, $c \in \{1, \ldots, 18\}$, ein negativer Vergleichswert für das entsprechende Attribut.

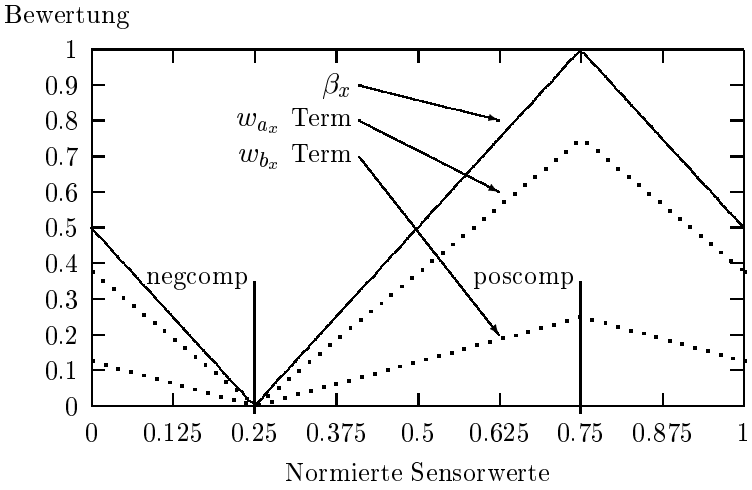

Abbildung 4.7: Verlauf der quasi linearen Klassifikationsfunktion 4.14 für die Gewichte $w_a = 0.75$ und $w_b = 0.25$.

In Abbildung 4.7 wird der Verlauf dieser β_x Definition grafisch dargestellt. Als Ergebnis liefert der Handformaktuator ein 5-Tupel, in welchem jeder Wert in einem **Skalenfeld** zur Verfügung gestellt wird. Diese Art Felder liefern normierte Werte zwischen 0 und 1, welche den *Grad* des zu bewertenden

Attributs widerspiegeln. Typischerweise können so relative Positionen eines Körperteils zu einem anderen oder Auslenkungen eines Körpergelenks repräsentiert werden, für den Handformaktuator sind dieses die Streckungswerte der einzelnen Finger:

$$FM(t) = \left\{ \begin{array}{|c|c|c|c|c|} \hline \texttt{AlignThumb} & \texttt{AlignIndex} & \texttt{AlignMiddle} & \texttt{AlignRing} & \texttt{AlignPinkie} \\ \hline \beta(t'_0, \dots)[1] & \beta(t'_0, \dots)[2] & \beta(t'_0, \dots)[3] & \beta(t'_0, \dots)[4] & \beta(t'_0, \dots)[5] \\ \hline \multicolumn{5}{c}{\vdots} \\ \hline \beta(t'_m, \dots)[1] & \beta(t'_m, \dots)[2] & \beta(t'_m, \dots)[3] & \beta(t'_m, \dots)[4] & \beta(t'_m, \dots)[5] \\ \hline \end{array} \right.$$

Dabei sei $t'_m = min(\sigma(t^e_{y-1}, \dots) \cap \sigma(t^e_y, \dots))$ (n. Definition 9 auf S. 90).

Diese Feldinformationen des Handformaktuators stehen nun weiteren Verarbeitungsstufen im Erkennungs- und Auswertungsprozess zur Verfügung.

Ein NDOF-Bewegungsaktuator

Sei $e_1 = (S_1(t), \vec{\kappa}_1(t))$ ein Eingabekanal, welcher Positions- und Ausrichtungsinformationen abstrahiert, wie sie beispielsweise durch einen VR-typischen elektromagnetischen Positions- und Orientierungssensor geliefert werden:

$$\vec{\kappa}_1 : \mathbb{R} \mapsto \mathcal{M}(4, 4, \mathbb{R})$$

$$\mathcal{M}(4, 4, \mathbb{R}) = \left\{ \begin{pmatrix} a_{11} & \dots & a_{14} \\ \vdots & & \vdots \\ a_{41} & \dots & a_{44} \end{pmatrix} \mid a_{xy} \in \mathbb{R} \right\} \tag{4.15}$$

$\vec{\kappa}_1$ liefert homogene Transformationsmatrizen, Elemente aus $M(4, 4, \mathbb{R})$. Dieses ist von der Sensorik nicht zwingend vorgegeben[11], aber es bildet für die benötigten Operationen sowie die Einbindung in das VR-System ein einheitliches Format. Eine Umrechnung in andere Darstellungsformate, z.Bsp. die Repräsentation von Rotationsanteilen mittels Quaternionen, kann beispielsweise für eine Interpolation wünschenswert sein und ist jederzeit möglich. Die Normierungsfunktion transformiert diese Daten in ein gemeinsames Basiskoordinatensystem:

$$\nu_1 : \mathcal{M}(4, 4, \mathbb{R}) \mapsto \mathcal{M}(4, 4, \mathbb{R}) \tag{4.16}$$

$$\nu_1(\vec{\kappa}_1(t)) = T_{c1} \cdot T_r \cdot \vec{\kappa}_1(t) \cdot T_t \cdot T_{c2} \cdot {}_pT^{root} \tag{4.17}$$

[11] Die Sensorik lässt i.A. unterschiedliche Ausgabeformate zu, z.Bsp. Position oder Orientierung alleine bzw. kombiniert etc.

Dabei seien

- T_{c1}, T_{c2} Anknüpfungspunkte für zusätzliche Kalibrationen,
- T_r, T_t Abbildungstransformationen zwischen Sensor- und Szenenbezugssystem,
- ${}_pT^{root}$ die Gesamttransformation vom Vaterknoten des Aktuators zum Szenenbasisknoten (root-node).

ν_1 veranschaulicht die Einbindung in das Szenenreferenzsystem durch die Einbettung in den Szenengraphen und die damit verbundenen Ausprägungen der einzelnen T_x, insbesondere von ${}_pT^{root}$ (s. dazu auch Abbildung 4.6 auf S. 91). Die T_c abstrahieren dabei vom eigentlichen Befestigungspunkt bei den häufig Verwendung findenden exoskeletalen Sensoren. Werden diese beispielsweise auf einem Arm befestigt, soll dagegen aber der *Mittelpunkt* des Arms gemessen werden, so kann die gewünschte Position durch die Festlegung von

$$T_{c_1} = MakeTransMat^{12}(0, 0, (-0.5) * \text{Armdicke}) \tag{4.18}$$

erreicht werden. Dabei verläuft die z-Achse des Sensors in negativer Richtung durch den Körperbefestigungspunkt. T_r und T_t korrigieren etwaige Unterschiede zwischen dem Empfänger- und Senderreferenzsystem des Sensors respektive und der virtuellen Szene. Auch in diesem Beispiel handelt es sich um einen Monokanal-Aktuator. Es hat sich gezeigt, dass eine Standardform dieser Sensorik — basierend auf dem elektromagnetischem Prinzip — unzuverlässig in Bezug auf eine feste Taktrate ist. Daher wird eine Kopplung über die Synchronisationsfunktion σ_2 etabliert:

$$\sigma_2(\mathbb{R} \times \mathcal{P}(\mathbb{R} \mapsto \mathcal{P}(\mathbb{R})) \times \mathbb{R}) \mapsto \mathcal{P}(\mathbb{R}) \tag{4.19}$$

$$\begin{gathered} \sigma_2(t, \{T_1(t)\}, i) = \\ \{t'_x \mid t^{min}(t) \leq t'_x \leq t^{max}(t) \wedge \forall x > k : t'_x - t'_{x-1} = i, \text{ mit } i \in \mathbb{N}\} \end{gathered} \tag{4.20}$$

Dabei sei $t'_k = t^{min} = t^1_k$, das heißt die Intervalle beginnen an der unteren Schranke; i wird auf eine angenommene maximale Zeitspanne zwischen zwei Samplezeitpunkten gesetzt. Ein NDOF-Bewegungsaktuator dieser Form kann sehr verschiedene Informationen als Ergebnis von β liefern.

Eine der einfachsten Möglichkeit besteht in der Berechnung eines einzelnen Punktes, z.Bsp. als Basis einer Trajektorienverfolgung oder -analyse:

$$\beta_1 : \mathbb{R} \mapsto \mathbb{R}^3 \tag{4.21}$$

$$\beta_1(t) = (0,0,0) \cdot \vec{\nu}_1(\vec{\kappa}_1(t) \tag{4.22}$$

Zusätzlich kann über die vorhandene zeitabhängige Komponente in den Kanälen die mittlere Dynamik der Bewegung über einem angegebene Zeitintervall berechnet werden. So ergibt sich die mittlere Geschwindigkeit im Intervall $[t-\epsilon, t]$ durch:

$$\beta_2 : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R} \tag{4.23}$$

$$\beta_2(t,\epsilon) = \frac{\|(0,0,0) \cdot \vec{\nu}_1(\vec{\kappa}_1(t)) - (0,0,0) \cdot \vec{\nu}_1(\vec{\kappa}_1(t-\epsilon))\|}{\epsilon} \tag{4.24}$$

Und darauf basierend die mittlere Beschleunigung:

$$\beta_3 : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R} \tag{4.25}$$

$$\beta_3(t,\epsilon) = \frac{\beta_2(t,\epsilon) - \beta_2(t-\epsilon,\epsilon)}{\epsilon} \tag{4.26}$$

Weiterhin definieren die durchschrittenen Punkte fortlaufende Bahnsegmente, wobei je zwei adjazente Segmente a) durch die Positionsänderungen eingeschlossene Winkel definieren und b) darüber hinaus im Falle nicht paralleler Segmente eine Ebene bestimmen. Diese Informationen werden bereits auf der Aktuatorbasis berechnet, um darauf aufsetzend später Regularitäten zu finden und so ähnliche Trajektorienteilstücke zu bestimmen. Für den Wert der Winkelveränderung gilt:

$$\beta_4 : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R} \tag{4.27}$$

$$\beta_4(t,\epsilon) = 180^\circ - \arccos\left(\frac{(\beta_1(t-\epsilon) - \beta_1(t-2\epsilon)) \cdot (\beta_1(t) - \beta_1(t-\epsilon))}{\|(\beta_1(t-\epsilon) - \beta_1(t-2\epsilon))\| * \|(\beta_1(t) - \beta_1(t-\epsilon))\|}\right) \tag{4.28}$$

Gemäß dieser Winkelwerte wird die Geschwindigkeit der Winkelveränderung (z.Bsp. in °/sec) und die entsprechende Beschleunigung bestimmt:

$$\beta_5 : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R} \tag{4.29}$$

$$\beta_5(t,\epsilon) = \frac{(\beta_4(t,\epsilon) - \beta_4(t-\epsilon,\epsilon))}{\epsilon} \tag{4.30}$$

$$\beta_6 : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R} \tag{4.31}$$

$$\beta_6(t,\epsilon) = \frac{(\beta_5(t,\epsilon) - \beta_5(t-\epsilon,\epsilon))}{\epsilon} \tag{4.32}$$

Die resultierende Ebene zweier Segmente ist gegeben durch einen Punkt und die entsprechende Normale:

$$\beta_7 : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}^3 \tag{4.33}$$

$$\beta_7(t, \epsilon) = (\beta_1(t - \epsilon) - \beta_1(t - 2\epsilon)) \times (\beta_1(t) - \beta_1(t - \epsilon)) \tag{4.34}$$

Auch für das Ergebnis von β_7 lassen sich der Winkel zwischen zwei aufeinanderfolgenden Normalen durch eine Funktion β_8, die Winkeländerungsgeschwindigkeit durch β_9 und die entsprechende Beschleunigung durch eine Funktion β_{10} berechnen. Diese werden hier nicht weiter erläutert, da sie analog zu den bereits definierten Funktionen (nur auf den Basiswerten von β_7) operieren.

Daneben verankern die Aktuatoren Körperteil-entsprechende Referenzstrahlen (wie in Abbildung 4.1 auf S. 69). Diese Informationen werden durch folgende Funktionen ermittelt. Verlaufe die y-Achse des Sensors parallel zum Arm und positiv in Richtung der Hand, so berechnet sich der primäre Referenzstrahl[13] des Arms nach:

$$\beta_{11} : \mathbb{R} \mapsto \{(x, y) \mid x, y \in \mathbb{R}^3\} \tag{4.35}$$

$$\beta_{11}(t) = ((0, 0, 0) \cdot \vec{\nu}_1(\vec{\kappa}_1(t)), (0, 1, 0) \cdot \vec{\nu}_1(\vec{\kappa}_1(t))) \tag{4.36}$$

Die Gesamtberechnungsfunktion β fasst diese Funktionen anschließend zusammen:

$$\beta : \mathbb{R} \mapsto (\mathcal{A}_1 \times \cdots \times \mathcal{A}_{10} \times (\mathcal{A}_{11} \times \mathcal{A}_{12})) \tag{4.37}$$

$$\begin{aligned} \beta(t) = (&\beta_1(t), \beta_2(t, \epsilon), \beta_3(t, \epsilon), \beta_4(t, \epsilon), \beta_5(t, \epsilon), \\ &\beta_6(t, \epsilon), \beta_7(t, \epsilon), \beta_8(t, \epsilon), \beta_9(t), \beta_{10}(t), \beta_{11}(t)) \end{aligned} \tag{4.38}$$

Die ϵ-Komponente wird dabei als ein konstanter Faktor angenommen. Sinnvolle Werte für deren Festlegung sind bei gegebenem σ_2 beispielsweise die Abstände der einzelnen t'_x. Die Ergebnisse der einzelnen Berechnungen repräsentieren aktuelle Werte für den Zeitpunkt t für die Attributsequenzen und werden bezogen auf den Aktuatorknoten in verschiedenen **Vektorfeldern** weitergegeben. Das `Position`-Feld liefert beispielsweise Stützpunkte im *Szenen*raum der während der Bewegung durchschrittenen Kurve des entsprechenden Punktes (s. dazu Abbildung 4.8 auf S. 102). `PrimFrom` und `PrimTo` liefern entsprechende Richtungsinformationen. `AvgSpeed` und

[13] Der Strahl wird als Basispunkt und Richtungsvektor notiert.

`AvgAccel` liefern die Geschwindigkeit und die Beschleunigung des Punktes `PrimFrom` respektive. Für die übrigen Werte werden entsprechende Kürzel verwendet:

$$FM(t) = \left\{ \begin{array}{cccc} \texttt{Position} & \texttt{AvgSpeed} & \texttt{AvgAccel} & \texttt{SegAng} \\ \beta(t'_0)[1] & \beta(t'_0)[2] & \beta(t'_0)[3] & \beta(t'_0)[4] \\ & & \vdots & \\ \beta(t'_m)[1] & \beta(t'_m)[2] & \beta(t'_m)[3] & \beta(t'_m)[4] \end{array} \right. \bullet\bullet\bullet$$

$$\bullet\bullet\bullet \begin{array}{cccc} \texttt{SegAngSpeed} & \texttt{SegAngAccel} & \texttt{Norm} & \texttt{NormAng} \\ \beta(t'_0)[5] & \beta(t'_0)[6] & \beta(t'_0)[7] & \beta(t'_0)[8] \\ & & \vdots & \\ \beta(t'_m)[5] & \beta(t'_m)[6] & \beta(t'_m)[7] & \beta(t'_m)[8] \end{array} \bullet\bullet\bullet$$

$$\bullet\bullet\bullet \begin{array}{cccc} \texttt{NormAngSpeed} & \texttt{NormAngAccel} & \texttt{PrimFrom} & \texttt{PrimTo} \\ \beta(t'_0)[9] & \beta(t'_0[10] & \beta(t'_0)[11][1] & \beta(t'_0)[11][2] \\ & & \vdots & \\ \beta(t'_m)[9] & \beta(t'_m)[10] & \beta(t'_m)[11][1] & \beta(t'_m)[11][2] \end{array}$$

In Abbildung 4.8 wird allein die Transformationswirkung des über die Ausgabefelder etablierten NDOF-Bewegungsaktuators durch das `Position`-Feld auf die Sensordaten im Kontext der aktuellen VR-Szene illustriert.

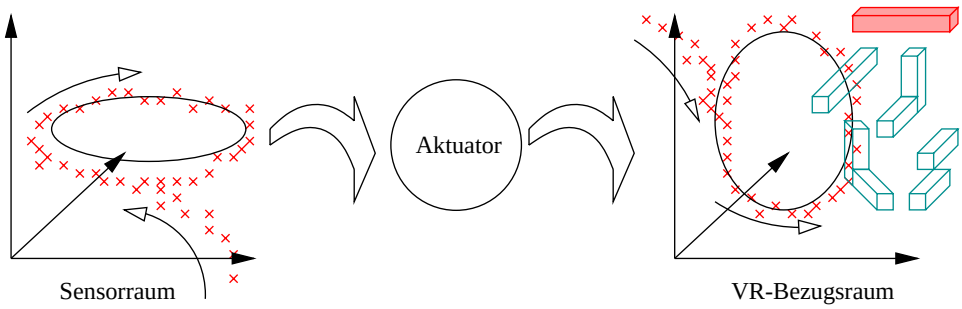

Abbildung 4.8: Transformationswirkung eines NDOF-Bewegungsaktuators über ein Vektorfeld in Bezug auf gegebene Basissysteme.

Ein Multikanal-NDOF-Bewegungsaktuator

Im folgenden Abschnitt wird das Konzept eines Mehrkanalaktuators zur gekoppelten Deixisauswertung über Zeige- und Blickrichtung entwickelt. Das Ziel ist es, aus den sensorischen Daten möglichst aussagekräftige *spatiale Referenzinformationen* zu ermitteln. Diese sollen einerseits als Grundlage einer Auswertung bedeuteter Orte und Positionen an sich zur Verfügung stehen. Andererseits sollen sie dazu dienen, den Benutzer umgebenden Raum insofern zu partitionieren, als dass darin befindliche Objekte durch ihre deixisrelative Lage einer Ordnung unterliegen, welche die Wahrscheinlichkeit widerspiegelt, inwieweit diese Objekte referenziert wurden. Betrachtet wird in diesem Beispiel allein die Richtungs- und Tiefeninformation. Ausgabeziel ist ein Referenzbasispunkt $\vec{r}$, eine Referenzrichtung $\vec{d}$, ein darauf liegender Ordnungsbasispunkt $\vec{o}$ sowie eine Abschätzung, wie hoch die Wahrscheinlichkeit eines Zeigens aufgrund der unterschiedlichen Blick- und Zeigerichtung ist.

Seien $e_1 = (S_1(t), \vec{\kappa}_1(t))$, $e_2 = (S_2(t), \vec{\kappa}_2(t))$ zwei Eingabekanäle , welche wiederum Positions- und Ausrichtungsinformationen abstrahieren:

$$\vec{\kappa}_1, \vec{\kappa}_1 : \mathbb{R} \mapsto \mathcal{M}(4, 4, \mathbb{R}) \quad \text{(s. (4.15))} \tag{4.39}$$

Die Normierungsfunktionen transformieren diese Daten in das gemeinsame Basiskoordinatensystem (*root* der virtuellen Szene):

$$\nu_1, \nu_2 : \mathcal{M}(4, 4, \mathbb{R}) \mapsto \mathcal{M}(4, 4, \mathbb{R}) \tag{4.40}$$

Die einzelnen Transformationen T_x unterscheiden sich, da unterschiedliche Extremitäten gemessen werden, in diesem Fall die Blickrichtung und eine Zeigerichtung. Angenommen wird eine Befestigung des Blickrichtungsensors an einer VR-typischen Shutterbrille zur 3D-Bildgenerierung und des Zeigerichtungssensors am Handgelenk des Benutzers. Da es sich aber um eine einzige Knoteninstanz handelt, existiert nur ein ${}_pT^{root}$. Unter der Annahme, dass die Sensordaten bezüglich einer einheitlichen Basis vorliegen, ist zusätzlich noch T_t identisch. Zur Erläuterung der übrigen T_x wird auf Abschnitt 4.2.4 verwiesen:

$$\begin{aligned} \nu_1(\vec{\kappa}_1(t)) &= T_{c1_1} \cdot T_{r_1} \cdot \vec{\kappa}_1(t) \cdot T_t \cdot T_{c2_1} \cdot {}_pT^{root} \\ \nu_2(\vec{\kappa}_2(t)) &= T_{c1_2} \cdot T_{r_2} \cdot \vec{\kappa}_2(t) \cdot T_t \cdot T_{c2_2} \cdot {}_pT^{root} \end{aligned} \tag{4.41}$$

ν_1 liefert die Transformation eines Referenzkoordinatensystems, welches zwischen den Augen des Benutzers lokalisiert ist und dessen Primärstrahl (gege-

ben durch den 0-Punkt des Koordinatensystems und die y-Achse) in Blickrichtung verläuft (s. zur Erläuterung Abbildung 4.1 auf S. 69). ν_2 abstrahiert die Position der Zeigefingerspitze (im Gegensatz zur Handwurzel in Abbildung 4.1) und als Ausrichtung der Primärachse die Verlängerung des rechten Unterarms. Die Auswertung der mit diesem Aktuator gelieferten Daten erfolgt in enger Kopplung zur dargestellten Szene, da diese a) die zu ordnenden Objekte zur Verfügung stellt und b) die aktuelle Szenenprojektion reflektiert. Daher wird für diesen Aktuator eine Auswertung zu jedem Frame durchgeführt. Als Synchronisationsfunktion wird σ_1 angewendet und die Aktuatorfunktion zum Zeitpunkt t^{max} getriggert:

$$\sigma_1 : (\mathbb{R} \times \mathcal{P}(\mathbb{R} \mapsto \mathcal{P}(\mathbb{R}))) \mapsto \mathcal{P}(\mathbb{R}) \tag{4.42}$$

$$\sigma_1(t, \{T_1(t), \ldots, T_i(t)\}) = \{t' \mid t^{min}(t) \leq t' \leq t^{max}(t)\} \tag{4.43}$$

Die Berechnungsfunktion implementiert eine Heuristik, um formabhängige Eigenschaften des menschlichen Zeigens zu berücksichtigen. Dabei werden an dieser Stelle folgende Annahmen über den Zusammenhang zwischen gestischer Ausführung und Verlässlichkeit der Zeigerichtung und des Zeigeortes zugrunde gelegt:

Je expressiver eine Zeigegeste in Bezug auf ihre spatiale Ausdehnung ausgeführt wird, also je mehr der zeigende Arm elongiert wird, um so genauer wird das Ziel über die Armrichtung angezeigt. Dabei resultiert die gesuchte Richtung sowohl aus der Zeige- als auch aus der Blickrichtung. Im Idealfall kreuzen sich beide am *Point of Interest* (POI). Liegen Blick- und Zeigerichtung zu weit auseinander, so wird die Zeigerichtung unzuverlässig. Liegt der POI zu nah am Zeigebasispunkt, so entspricht das Zeigen mehr einem *Peilen* über die Zeigespitze (in diesem Fall über die Fingerspitze) hinweg. Die geschilderten Eigenschaften werden durch β umgesetzt.

$$\beta : \mathbb{R} \mapsto (\mathbb{R} \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3) \tag{4.44}$$

$$\begin{aligned} \beta(t) = (&\beta_1(\vec{\nu}_1(\vec{\kappa}_1(t)), \vec{\nu}_2(\vec{\kappa}_2(t))), \\ &\beta_2(\vec{\nu}_1(\vec{\kappa}_1(t)), \vec{\nu}_2(\vec{\kappa}_2(t))), \\ &\beta_3(\vec{\nu}_1(\vec{\kappa}_1(t)), \vec{\nu}_2(\vec{\kappa}_2(t))), \\ &\beta_4(\vec{\nu}_1(\vec{\kappa}_1(t)), \vec{\nu}_2(\vec{\kappa}_2(t)))) \end{aligned} \tag{4.45}$$

Seien die y-Achsen parallel zu den primären Richtungsstrahlen der Extremitäten. Dann sind durch je zwei Punktvektoren Referenzstrahlen folgendermaßen gegeben:

- Blickbasis (gaze-base) $\vec{g}_b = (0,0,0) * \vec{\nu}_1(\vec{\kappa}_1(t))$
- Zeigebasis (pointing-base) $\vec{p}_b = (0,0,0) * \vec{\nu}_2(\vec{\kappa}_2(t))$
- Blickrichtung (gaze-direction) $\vec{g}_d = (0,1,0) * \vec{\nu}_1(\vec{\kappa}_1(t))$
- Zeigerichtung (pointing-direction) $\vec{p}_d = (0,1,0) * \vec{\nu}_2(\vec{\kappa}_2(t))$.

So bewertet β_1 die ermittelten Daten gemäß des Wertes des eingeschlossenen Winkels:

$$\beta_1(\vec{\nu}_1(\vec{\kappa}_1(t)), \vec{\nu}_2(\vec{\kappa}_2(t))) = 0.5 * (1 + tanh(s * (\vec{g}_d \cdot \vec{p}_d - w))) \tag{4.46}$$

Hierbei gibt der Parameter s die Steilheit und der Parameter w den Wendepunkt der Bewertungskurve an. Für eine positive Bewertung (> 0.5) ab einem Grenzwinkel von ca. $53°$ wird $s = 20$ und $w = 0.6$ gesetzt. Die Abbildungen 4.9 und 4.10 geben den Verlauf für diese Parameter in Anhängigkeit der entsprechenden Winkel wieder.

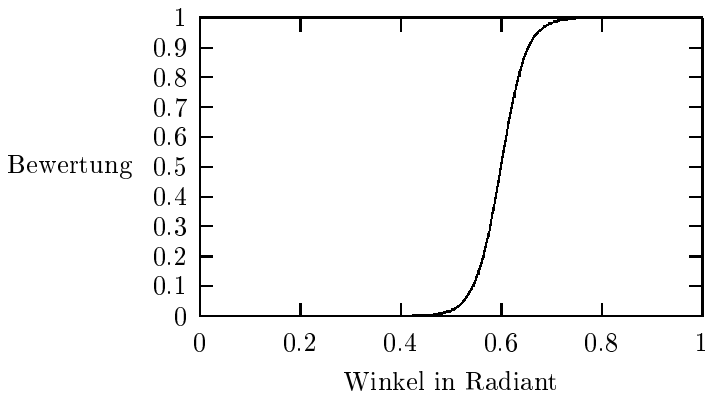

Abbildung 4.9: Verlauf der Bewertungsfunktion β_1 für $s = 20$ und $w = 0.6$.

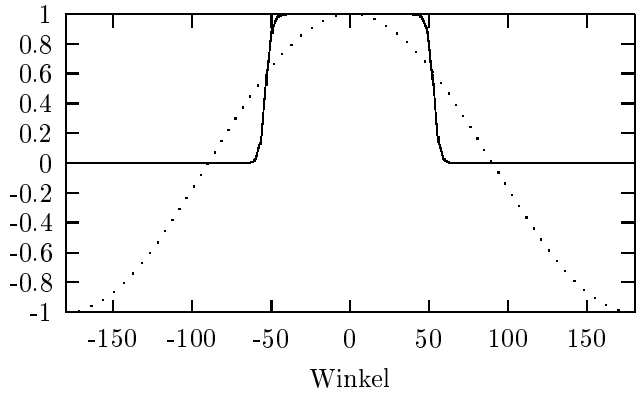

Abbildung 4.10: Bewertungsfunktion und Verlauf der Cosinusfunktion. Die gemeinsame Ordinate entspricht je nach Plot entweder dem Cosinus des Winkels (gepunktete Linie) oder der entsprechenden Bewertung durch β_1 (durchgezogene Linie).

Für die Bestimmung der weiteren Ausgabekomponenten werden einfache Fallunterscheidungen angewendet. Sei d die Distanz zwischen $\vec{p}_b$ und dem durch $\vec{g}_b, \vec{g}_d$ gegebenen Strahl und ϵ ein entsprechender Grenzwert. So ist die Wahl des Basispunktes abhängig vom Verhältnis d zu ϵ:

$$\beta_2(\vec{\nu}_1(\vec{\kappa}_1(t)), \vec{\nu}_2(\vec{\kappa}_2(t))) = \begin{cases} \vec{p}_b & \text{wenn } d > \epsilon \\ \vec{g}_b & \text{sonst.} \end{cases} \tag{4.47}$$

Für die Wahl des Richtungsvektor wird entsprechend gesetzt:

$$\beta_3(\vec{\nu}_1(\vec{\kappa}_1(t)), \vec{\nu}_2(\vec{\kappa}_2(t))) = \begin{cases} \vec{p}_d & \text{wenn } d > \epsilon \\ \vec{g}_d & \text{sonst.} \end{cases} \tag{4.48}$$

Der gesuchte Ordungsbasispunkt entspricht dem POI, also dem Punkte $\vec{s}$ auf dem Zeigestrahl, welcher minimalen Abstand zum Blickrichtungsstrahl besitzt (für auseinanderlaufende Strahlen wird dieses immer der Default $\vec{p}_b$ sein):

$$\beta_4(\vec{\nu}_1(\vec{\kappa}_1(t)), \vec{\nu}_2(\vec{\kappa}_2(t))) = \vec{s} \tag{4.49}$$

Ein Multikanal-NDOF-Symmetrieaktuator

Nachdem die grundlegende Vorgehensweise mit Hilfe der formalen Aktuatordefinition und den bereits erarbeiteten Aktuatortypen verdeutlicht wurde, wird hier abschließend ein Multikanal-NDOF-Aktuator zur Ermittlung der für die Erkennung symmetrischer Handbewegungen notwendigen Informationen konzipiert. Basis für diese Symmetriebestimmung sind die in Abschnitt 4.1.1 beschriebenen Bezugssysteme der Hände. Zur Andeutung des Prinzips werdens nur die Primär- und Sekundärstrahlen betrachtet.
$e_1 = (S_1(t), \vec{\kappa}_1(t))$, $e_2 = (S_2(t), \vec{\kappa}_2(t))$ sind zwei Eingabekanäle, welche wiederum Positions- und Ausrichtungsinformationen abstrahieren. Für die einzelnen Definitionen der ν_1, ν_2 sowie von σ wird auf den letzten Abschnitt verwiesen. An dieser Stelle werden allein die elementaren Berechnungsschritte und die entsprechenden Bezeichner der Attributsequenzen eingeführt.
`HandDistance`:

$$\beta_1(t) = \|(0,0,0) * \vec{\nu}_1(\vec{\kappa}_1(t)) - (0,0,0) * \vec{\nu}_2(\vec{\kappa}_2(t))\| \tag{4.50}$$

`AngPrimPrim`:

$$\beta_2(t) = \arccos\left(\frac{((0,1,0) * \vec{\nu}_1(\vec{\kappa}_1(t))) \cdot ((0,1,0) * \vec{\nu}_2(\vec{\kappa}_2(t)))}{\|((0,1,0) * \vec{\nu}_1(\vec{\kappa}_1(t)))\| * \|((0,1,0) * \vec{\nu}_2(\vec{\kappa}_2(t)))\|}\right) \tag{4.51}$$

`AngPrimSek`:

$$\beta_3(t) = \arccos\left(\frac{((0,1,0) * \vec{\nu}_1(\vec{\kappa}_1(t))) \cdot ((0,0,-1) * \vec{\nu}_2(\vec{\kappa}_2(t)))}{\|((0,1,0) * \vec{\nu}_1(\vec{\kappa}_1(t)))\| * \|((0,0,-1) * \vec{\nu}_2(\vec{\kappa}_2(t)))\|}\right) \tag{4.52}$$

`AngSekPrim`:

$$\beta_4(t) = \arccos\left(\frac{((0,0,-1) * \vec{\nu}_1(\vec{\kappa}_1(t))) \cdot ((0,1,0) * \vec{\nu}_2(\vec{\kappa}_2(t)))}{\|((0,0,-1) * \vec{\nu}_1(\vec{\kappa}_1(t)))\| * \|((0,1,0) * \vec{\nu}_2(\vec{\kappa}_2(t)))\|}\right) \tag{4.53}$$

`AngSekSek`:

$$\beta_5(t) = \arccos\left(\frac{((0,0,-1) * \vec{\nu}_1(\vec{\kappa}_1(t))) \cdot ((0,0,-1) * \vec{\nu}_2(\vec{\kappa}_2(t)))}{\|((0,0,-1) * \vec{\nu}_1(\vec{\kappa}_1(t)))\| * \|((0,0,-1) * \vec{\nu}_2(\vec{\kappa}_2(t)))\|}\right) \tag{4.54}$$

Auf diesen elementaren Berechnungen werden analog zu (4.30) und (4.34) über einem Intervall ϵ entsprechende Differenzen für die Bestimmung der Geschwindigkeit `HandDistanceSpeed`/`AngXXSpeed`[14] und Beschleunigung `HandDistanceAccel`/`AngXXAccel` der Parameteränderung berechnet.

[14] `XX` steht für die entsprechenden `PrimPrim`, `PrimSek`, ...

4.2.5 Aktuatoren vs. Knoteninstanzen

Die Differenzierung zwischen Abstrakten Aktuatoren und Aktuatorenknoten in einer szenengraphbasierten Modellierung erlaubt die notwendige Trennung der in der jeweiligen Bearbeitungsphase zu betrachtenden Teilaufgaben. Abstrakte Aktuatoren kapseln die Sensorik abhängig von deren unterschiedlichem Sampleverhalten und definieren die benütigten Normierungs- und Berechnungsfunktionen. Aktuatorknoten dagegen spezifizieren a) die lose Kopplung der Sensorauswertung in der geschlossenen Render-Loop und b) notwendige Transformationsdaten für $\mathbb{R}^3$ basierte Sensorinformationen und konkretisieren damit deren ν und β. In Abbildung 4.11 wird der Zusammenhang zwischen den Abstrakten Aktuatoren und den Knoteninstanzen verdeutlicht.

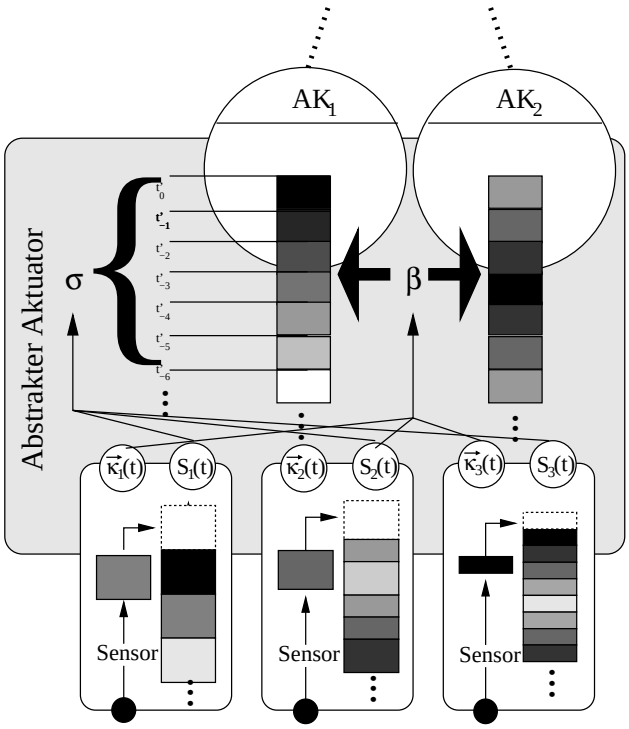

Abbildung 4.11: Zusammenspiel zwischen Abstraktem Aktuator und Aktuatorknoten. Zwei Aktuatorknoten (AK_1 und AK_2) referenzieren einen Abstrakten Aktuator mit drei Kanälen. Die Attributsequenzen werden in einer einheitlichen zeitlichen Auflösung gemäß σ berechnet.

Betrachtet man beide Konzepte vom Standpunkt einer zu entwickelnden Anwendung, so wird es — wie schnell einsichtig wird — pro Anwendung *einen* Abstrakten Aktuator geben und dagegen *mehrere* Aktuatorknoten, welche diesen referenzieren. Durch die ${}_2A$ Komponente in der formalen Definition des Aktuatorknotens wird diese 1:N-Modellierung bereits vorgesehen. So werden Attributsequenzen mit einer einheitlichen Auflösung und gleichen Zeitpunkten generiert. Diese Attributsequenzen bilden, wie in den folgenden Abschnitten erläutert wird, die Basis der Gestenerkennung.

4.3 Gestenerkennung mit PrOSA

Die entwickelten Aktuatorkonzepte gewährleisten es, von unterschiedlichen Sensoriken zu abstrahieren, und stellen bedeutsame Information über die gestische Bewegung eines Anwenders in einer einheitlichen Bezugsbasis und zeitlichen Auflösung im Kontext von VR-Anwendungen zur Verfügung. In diesem Abschnitt werden auf Basis der Aktuatoren und den Attributsequenzen benötigte Komponenten und Verfahren ausgearbeitet, um die Erkennung und Analyse menschlicher Gestik in einem VR-System zu gewährleisten. Ziel ist es, auf Basis der spatiotemporalen Expressionsmerkmale (s. Abschnitt 4.1.2) die Erkennung und Auswertung der folgenden Interaktionsgestentypen (s. Abschnitt 3.1) zu ermöglichen: Deiktische, kinemimische/mimetische und ikonische (spatiographische/pictomimische) Gesten. Bei den dabei Verwendung findenden Verfahren wird besonderes Augenmerk auf a) eine geeignete Parametrisierung und b) die Möglichkeit der synchronisierten Ausführung mit einer externen Quelle gelegt, um so die Anbindung an ein VR-System bereitzustellen.

Die Gesamtheit der zum Gelingen dieses Vorhabens bereits vorgestellten sowie der hier noch ausgearbeiteten Konzepte wird mit dem Akronym *PrOSA* (Patterns On Sequences of Attributes) bezeichnet. Diese Namensgebung basiert auf folgenden zwei zentralen Vorgehensweisen: Zum einen auf der Verankerung der Gestenerkennung und -analyse auf den durch die Abstrakten Aktuatoren vorgegebenen Berechnungswerten ($\simeq$ Attributes) zu den Zeitpunkten aus σ ($\simeq$ Sequences) und den daraus resultierenden Attributsequenzen $A(t)$, vermittelt durch die $FM(t)$ der Aktuatorknoten, zum anderen auf der Verfahrensweise einer Suche nach Mustern ($\simeq$ Patterns) auf diesen Sequenzen.

Es wurde im Laufe der Arbeit bereits an entsprechenden Stellen zwischen Gestenerkennung und -analyse unterschieden. Diese Trennung soll an dieser Stelle noch einmal rekapituliert werden. Es bedeuten:

- **Gestenanalyse:** Feststellung der für das entsprechende Bewegungsmuster markanten Parameter.
- **Gestenerkennung:** Feststellung welche Art von Gestik ausgeführt wird, eine Klassifikationsleistung.

Nach der Diskussion über maschinelle Verfahren für Erkennungsaufgaben wird hier für die Lösung der sich stellenden Aufgabe — der Gestenerkennung

und der -analyse — ein hybrider Ansatz gewählt. Die grundlegenden analytischen Vorverarbeitungen werden auf der Aktuatorebene vollzogen. Für die Klassifikationsleistung — die Erkennung — findet bewusst eine Suche nach vorher definierten Mustern Verwendung, ein merkmals- oder eigenschaftsbasiertes Vorgehen. Das Wissen über die eine bestimmte Geste ausmachende Kombination von Merkmalen/Eigenschaften (properties) wird im Voraus als eine logische Verknüpfung von Tests explizit gemacht. Diese basieren entweder auf den vorher analytisch gewonnenen Bewegungsdaten der Aktuatoren oder auf Ergebnissen aus analogen Tests und Bewertungen. Im letzteren Fall resultiert dieses in einer hierarchischen Auswertung.

Zur Erläuterung dieses Ansatzes wird auf eine prädikatenlogische Darstellung zurückgegriffen. Die dabei Verwendung findenden Terme sind:

- Geschlossene Zeitintervalle $i_x : i_x = [t_{a_x}, t_{e_x}], \forall t_y \in i_x : t_y \in \sigma$ auf den Auswertezeitpunkten, gegeben durch Anfang und Ende des Intervalls.
- Properties p_x, elementare Tests, welche prüfen, ob die entsprechende Eigenschaft für jeden Zeitpunkt in einem Intervall gilt oder nicht gilt.

Mit diesen Termen und einem Prädikat HOLDS?, kann die Gültigkeit einer Testbedingung über einem Intervall i denotiert werden. Sei beispielsweise ArmMoving ein Property, welches besagt, ob sich der entsprechende Arm in Bewegung findet. Dann wird HOLDS? für dieses Property definiert als:

$$\text{HOLDS?(ArmMoving,i)} = \begin{cases} \text{true} & \text{wenn } \forall\, t_x \in i : \vec{a}_{\texttt{AvgSpeed}}(t_x) > 0.05 \\ \text{false} & \text{sonst.} \end{cases} \tag{4.55}$$

Dabei sei $\vec{a}_{AvgSpeed}(t_x)$ der Wert der durch `AvgSpeed` bezeichneten Attributsequenz zum Zeitpunkt t_x. Basiert ein Property auf einem Schwellwerttest (Thresholdtest) der Attributsequenz-Zugriffsfunktion (wie in (4.55)), wird im Folgenden auch äquivalent die Notation HOLDS?((`AvgSpeed` > 0.05),i) verwendet.

Greifgestenerkennung

Die logischen Verknüpfungen von Prädikaten, welche auf elementaren Propertytests basieren, dienen als *Templates*. Ein einfaches Beispiel soll die Form der Gestentemplatedefinition verdeutlichen. Die Bestimmung, ob eine Hand geschlossen ist, also der Benutzer zugreift (grasp), benötigt vier Propertytests, welche wiederum auf den Aktuatorinformationen des Handformaktuators aus Abschnitt 4.2.4 basieren:

$$\begin{aligned}\text{HOLDS?}(\texttt{Grasp}, i) = {} & \text{HOLDS?}((\texttt{AlignIndex} < 0.2), i_1) \\ & \wedge \text{HOLDS?}((\texttt{AlignMiddle} < 0.2), i_1) \\ & \wedge \text{HOLDS?}((\texttt{AlignRing} < 0.2), i_1) \\ & \wedge \text{HOLDS?}((\texttt{AlignPinkie} < 0.2), i_1)\end{aligned} \tag{4.56}$$

Diese Definition veranschaulicht an einem ersten Beispiel, in welcher Form eine Klassifikation mit Hilfe der Erkennung spatiotemporaler Gestenmerkmale (n. Abschnitt 4.1.2) ermöglicht wird: Durch die Propertytests wird im Greifgestenfall eine eindeutige **Postur** (Merkmal 3) erkannt und durch die explizite Verwendung von Zeitintervallen kann **Konstanz** (inhärent in Merkmal 4) in einem diskreten Propertytestergebnis ausgedrückt werden, wenn dieses nicht schon auf Aktuatorebene durch Differenzierungsschritte vorgesehen wird.

Victoryzeichenerkennung

Nimmt man — bezogen auf die Hände — zu den Streckungsinformationen der Finger auch die Fingerspreizung hinzu, lassen sich prinzipiell alle möglichen Arten von Handposturen mit Hilfe dieser Prädikatverknüpfungen beschreiben. Eine typische symbolische Geste — hier nur zur Demonstration betrachtet — ist das Victoryzeichen. Alle Finger sind dabei eingerollt, nur Zeige- und Mittelfinger sind gestreckt und abgespreizt. Dieses wird notiert als:

$$\begin{aligned}\text{HOLDS?}(\texttt{Victory}, i) = {} & \text{HOLDS?}((\texttt{AlignIndex} > 0.9), i_1) \\ & \wedge \text{HOLDS?}((\texttt{AlignMiddle} > 0.9), i_1) \\ & \wedge \text{HOLDS?}((\texttt{AlignRing} < 0.2), i_1) \\ & \wedge \text{HOLDS?}((\texttt{AlignPinkie} < 0.2), i_1) \\ & \wedge \text{HOLDS?}((\texttt{StiltedIndexMiddle} > 0.2), i_1)\end{aligned} \tag{4.57}$$

Kreisbahnerkennung

Merkmal 4 beschreibt Konstanz in einem Parameter als ein Anzeichen für ein primitives Bewegungsprofil. Dieses kann neben der hier eingeführten Intervallsicht schon auf Aktuatorebene durch eine entsprechende Differenzierung bereitgestellt werden. Für die Erkennung einer kreisförmigen Bewegung einer Hand wird dieses durch die Betrachtung der Dynamik von Winkeländerungen aufeinanderfolgender Segmente erreicht:

$$\begin{aligned}\text{HOLDS?}(\texttt{Orbitting}, i) = \text{HOLDS?}((\texttt{AvgSpeed} > 0.15), i_1)\\ \wedge \text{HOLDS?}((\texttt{SegAng} > 4), i_1)\\ \wedge \text{HOLDS?}((\texttt{SegAng} < 50), i_1)\\ \wedge \text{HOLDS?}((\texttt{SegAngSpeed} < 0.1), i_1)\\ \wedge \text{HOLDS?}((\texttt{NormAngSpeed} < 1), i_1)\\ \wedge \text{HOLDS?}((\texttt{NormAngAccel} < 0.1), i_1)\end{aligned} \tag{4.58}$$

4.3.1 Detektoren

Die Gestentemplates müssen in der laufenden Anwendung "quasi kontinuierlich"[15] auf ihre Erfüllung überprüft werden. Diese Funktion wird von Gestenerkennungs- und Analysemodulen — sogenannten **Detektoren** — in einem *Matching*-Vorgang ausgeführt. Detektoren kapseln diesen Matchingvorgang für alle neuen Sequenzwerte und dienen der Anbindung der Gestenerkennung an eine externe Synchronisationsinstanz, ähnlich der Möglichkeit der Anbindung von Aktuatorknoten an die Render-Loop.

Das Matchingresultat steuert zwei verschiedene Funktionen. Zum einen gewichtet eine resultierende Matchingbewertungsfunktion ω das Klassifikationsergebnis. Zum anderen dient eine Auswerte- und Analysefunktion α dazu, die *signifikanten gestischen Informationen* analog zum Bewertungsvorgang zu ermitteln. Beide Ergebnisse werden auf Detektorseite in zwei weiteren Attributsequenzen weitergegeben. Für die Ausgabewerte von ω wird dabei der Typ der Attributsequenzwerte eingeschränkt. Es gilt: $\omega : \mathbb{R} \mapsto \{x \mid x \in \mathbb{R} \wedge 0 \leq x \leq 1\}$. Diese Form wird hier als Signifikanzattributsequenz bezeichnet.

Im Gegensatz zu den objektiven Bewegungsdaten der Aktuatoren bewerten Detektoren subjektiv gemäß den jeweils implementierten Regeln. So

[15] Kontinuierlich im Sinne der zur Verfügung stehenden Bewegungsdatenauflösung in den Attributsequenzen.

ähneln sich beide Komponenten in der Verankerung etwaiger benötigter Testfunktionen. Dagegen obliegen die Synchronisations- und Normierungsaufgaben allein den Aktuatoren, um damit eine einheitliche Basis für die an dieser Stelle benötigte Analyse- und Klassifikationsaufgaben zu gewährleisten.

Definition 10 (Detektor) *Detektoren klassifizieren Bewegungsinformationen, indem sie Muster — logische Verknüpfungen von atomaren Testbedingungen — auf Bewegungsinformationen suchen. Als Resultat dieses Matchingvorgangs wird je ein Bewertungs- und ein Analysewert ermittelt. Ein Detektor ist ein 7-Tupel*

$$GD = (AI(t), \tau(t'), \omega(t'), \alpha(t'), T_{eval}(t), O(t), A(t)) \tag{4.59}$$

Dabei sei

- $AI(t)$ eine Menge von Attributsequenzen zum Zeitpunkt t
  $AI(t) = \{A_1(t), \ldots, A_i(t)\}$.
- $\tau(t')$ eine logische Verknüpfung von atomaren Testbedingungen (Properties) auf den $A_j(t)$, $t' \in T_{eval}(t)$,
  $\tau : \mathbb{R} \mapsto \{0, 1\}$.
- $\omega : \mathbb{R} \mapsto \{x \mid x \in \mathbb{R} \wedge 0 \leq x \leq 1\}$ eine Bewertungsfunktion, welche gemäß dem Resultat von τ ausgeführt wird und die Ergebnisse für O(t) liefert.
- $\alpha : \mathbb{R} \mapsto \mathcal{A}$ eine Analysefunktion, welche gemäß dem Resultat von τ ausgeführt wird und die Ergebnisse für A(t) liefert.
- $T_{eval} : \mathbb{R} \mapsto \mathcal{P}(\mathbb{R})$
  $T_{eval}(t) = \{t_k < t_{k+1} < \ldots < t_0 \leq t, t_i \in \mathbb{N}_0, k \in \mathbb{Z}_0^-\}$ eine zeitabhängig wachsende endliche Menge von diskreten, disjunkten und geordneten Auswertezeitpunkten.
- $O(t)$ eine Signifikanzattributsequenz zum Zeitpunkt t.
- $A(t)$ eine Attributsequenz zum Zeitpunkt t, welche ein Analyseergebnis repräsentiert.

Abbildung 4.12 (S. 116) illustriert das Zusammenspiel zwischen Aktuator und Detektor. Ein Aktuator gibt durch die Anzahl der pro Evaluationsschritt zu einem Zeitpunkt t gelieferten Feldwerte $FM(t)$ (basierend auf den $A(t)$ des Abstrakten Aktuators) eine einheitliche Auflösung der für die Detektorberechnung benötigten Eingangsdaten vor.

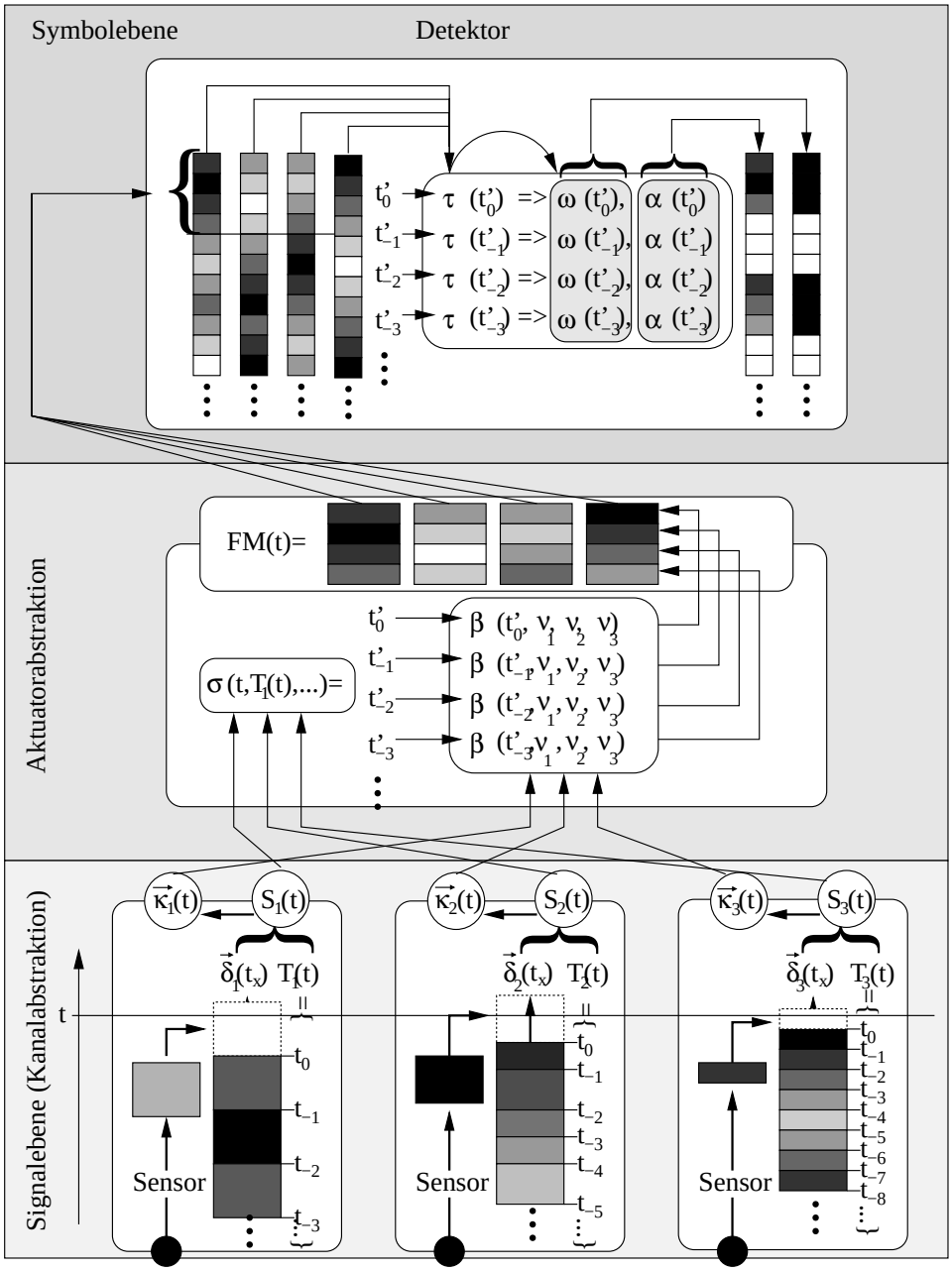

Abbildung 4.12: Singuläre Detektorauswertung an einem Aktuator mit drei Sensorquellen. Die pro Traversierungsschritt neu anfallenden Sensordaten werden nach der Bearbeitung eines Aktuators als synchronisierte Attributsequenzen an die Detektorebene weitergegeben.

Die unterschiedlichen Schattierungen der Attributsequenzenwerte in Abbildung 4.12 verdeutlichen die Art der Musterbildung. Die Festlegung der σ-Funktion und den damit gegebenen Auswertezeitpunkten resultiert in einer einheitlichen zeitlichen Auflösung, welche sich sukzessive von der Sensorebene bis auf die hier beschriebene Erkennerebene fortsetzt. Da Detektoren auf unterer Basis auf den Aktuatordaten operieren, ergibt sich natürlich für die Gestenerkennung und -analyse ebenfalls die entsprechende Latenzzeit der zugrunde liegenden Sensorik, da der früheste — im Sinne einer möglichen Interpolation auf den sensorischen Rohdaten — gültige Zeitpunkt durch t'_0 gemäß σ bestimmt ist.

4.3.2 Event- vs. Intervallergebnis

Die Bewertungsfunktion ω bestimmt, ob es sich bei den Signifikanzattributwerten um einfache Wahrheitwerte $\{0, 1\}$ handelt, also $\omega = \tau$ gesetzt wird, oder ob eine Gewichtung feinerer Granularität über $\mathbb{R}$ im angegeben Intervall eingesetzt wird. Dabei ist der Übergang von Ergebnissen aus $\mathbb{R}$ in den binären Wert immer durch das Hinzuziehen und den Vergleich des Berechnungsergebnisses mit einem oder mehreren vorher festgelegten Schwellwerten (Thresholds) zu realisieren. Der Einsatz dieser Form von Vergleichsbedingungen muss aber spätestens auf oberster Ebene, hier ist damit die Ebene der multimodalen Integration bezeichnet, die Klassifikationsleistung erbringen und damit die *Segmentierung* des Gestenstroms ermöglichen: Anhand einer Vergleichsschwelle muss eine Entscheidung gefällt werden ob die Geste vorliegt oder nicht. Daher gilt dann $\omega : \mathbb{R} \mapsto \{0, 1\}$. In Abbildung 4.12 wird dieses durch das Muster rein schwarzer oder weisser Attributwerte des Detektors in der ω-Ausgabesequenz verdeutlicht.
Bei den Sequenzwerten handelt es sich hier wiederum um diskretisierte Werte, allerdings mit einer einheitlichen Auflösung. Eine analoge Berechnung mit Hilfe einer geeigneten Interpolation wie über den sensorischen Eingabedaten wird an dieser Stelle nicht benötigt. Die rechnergestützte Auswertung wird letztendlich immer zu diskreten Zeitpunkten erfolgen. Hier wird daher angestrebt, eine möglichst *adäquate* Auflösung zu verwenden. Basierend auf der Arbeit von Lenzmann [54] wird eine minimale Diskretisierung im Bereich von 30msec gefordert, da dieser Wert als ein Minimalwert für das menschliche Auflösungsvermögen bei der Verarbeitung externer Signale anzunehmen ist und darüber hinaus auch nach (4.6) die sensorische Minimalrate übertrifft. Ist dieses bereits für die Festlegung von σ vorgesehen, so wird sich nach Spezifikation diese zeitliche Auflösung in den Attributsequenzen fortsetzen.

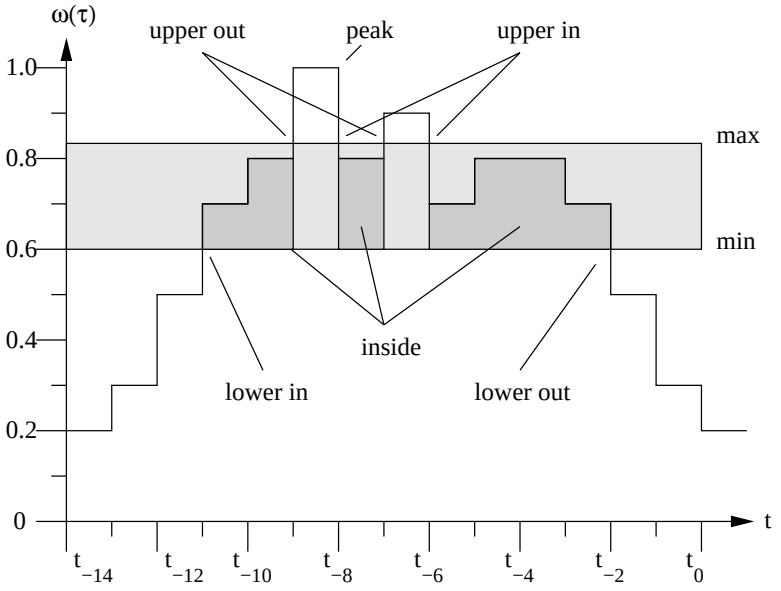

Abbildung 4.13: Mögliche Ereignisse bei Schwellwertüberprüfungen von Detektorergebnissen.

Die intervallbasierte Definition lässt noch Fragen offen, welche Entscheidungskriterien ein technisches System zur Grundlage hat, um weitere Operationen anzustoßen. Die Ausführung menschlicher Gestik wird prinzipiell in die Phasen **preparation**, *prestroke hold*, **stroke**, *poststroke hold* und **recovery/retract** (s. Abschnitt 2.2.1) gegliedert. Diese Phasen sind für Gesten unterschiedlich stark ausgeprägt. Die Stroke-Phase kann sich z.Bsp. über einen ganzen Zeitabschnitt erstrecken oder auf einen diskreten Zeitpunkt beziehen. Neben der intervallbasierten Sichtweise eines längeren Gültigkeitszeitraums eines Gestenstroke, können daher ebenso *Übergangsereignisse* der Sequenzwerte an den Thresholds die Auswertung eventbasierter Ereignisse zulassen. Beide Möglichkeiten werden hier berücksichtigt. Die Umsetzung des betrachteten Eventmodells wird im folgenden Abschnitt durch die Einführung einer temporalen Beziehung zwischen Intervallprädikaten (s. HOLDS?) erläutert.

Zeigegestenerkennung

Prinzipiell existieren für Tests gegen Thresholds unterschiedliche Zustände und Zustandsänderungen, welche eine Signalisierung einer Operation ermöglichen. Handelt es sich beispielsweise um einen Intervalltest wie er in Abbil-

dung 4.13 dargestellt wird, also um zwei Thresholds, welche eine untere und eine obere Schranke definieren, so kann es zu vier unterschiedlichen Überschreitungsformen in der ansteigenden oder in der absteigenden Flanke des Signals kommen. Je nach Konstellation von aktuellem und zurückliegendem Vergleichsergebnis wird dieses als *lower in* (Eintritt an unterer Schranke), *upper out* (Austritt an oberer Schranke), *upper in* (Eintritt an oberer Schranke) oder *lower out* (Austritt an unterer Schranke) notiert (s. Abbildung 4.13). Daneben ist nach wie vor das gesamte Intervall, also im Beispiel der Zustand *inside* oder bei nur einem Schwellwert die Zustände *over* oder *under* bezeichnend für ein Gestenmuster. Letztendlich ist auch der maximale Wert einer Berechnung, der *peak*, — sofern es einen solchen gibt — als wichtiges Datum von besonderem Interesse. Je nach Templatedefinition fließen entweder die als *atomar* zu betrachtenden Zustandsänderungen als *Events* in die Auswertung ein oder es wird über mehrere Schritte hinweg ein kontinuierlicher Vergleich mit *inside*-basierten Ergebnissen durchgeführt. Diese Informationen werden explizit durch die Ausgabeattributsequenz $O(t)$ in der Auflösung der Zeitschritte aus $T(t)$ bereitgestellt.

Um die Notation eventbasierter Übergangsereignisse zu ermöglichen, wird nach [2] ein weiteres Prädikat MEETS eingeführt, welches den zeitlichen Bezug zweier Zeitintervalle zueinander explizit macht:

$$\mathrm{MEETS}(i_2, i_1) = \begin{cases} \text{true} & \text{wenn } \forall\, t_x \in i_2 : t_x < t_{a_1} \\ & \qquad \wedge \nexists\, t_y : t_{e_2} < t_y < t_{a_1}, \\ \text{false} & \text{sonst.} \end{cases} \tag{4.60}$$

Mit diesem Prädikat lässt sich für die Detektion einer Zeigegeste das *peak* Event kenntlich machen, indem die Bewegungsstops der bewegenden Extremität als Übergang an einem Threshold definiert werden:

$$\begin{aligned} \mathrm{HOLDS?}(\texttt{Pointing}, i) = {} & \mathrm{HOLDS?}((\texttt{AlignIndex} > 0.8), i_1) \\ & \wedge \mathrm{HOLDS?}((\texttt{AlignMiddle} < 0.6), i_1) \\ & \wedge \mathrm{HOLDS?}((\texttt{AlignRing} < 0.5), i_1) \\ & \wedge \mathrm{HOLDS?}((\texttt{AlignPinkie} < 0.4), i_1) \\ & \wedge \mathrm{HOLDS?}((\texttt{AlignIndexSpeed} < 0.1), i_1) \\ & \wedge \mathrm{HOLDS?}((\texttt{AvgSpeed} < 0.1), i_1) \\ & \wedge \mathrm{HOLDS?}((\texttt{AvgSpeed} \geq 0.1), i_2) \\ & \wedge \mathrm{MEETS}(i_2, i_1) \end{aligned} \tag{4.61}$$

In diesem Beispiel fließt ein weiteres spatiotemporales Gestenmerkmal in den Erkennungsprozess ein: **Aktion und Pause** (Merkmal 1) — und die entsprechenden Übergänge zwischen diesen — segmentieren Gestenereignisse. Durch das Temporalprädikat MEETS wird dieses auf dem Bewegungsdatum explizit gemacht.

4.3.3 Verschaltungen von Detektoren

Die einheitliche Schnittstelle der Berechnungsergebnisse der Aktuatoren sowie der Detektoren ermöglichen es, die unterschiedlichen Berechnungsergebnisse beider Konzepte zu verknüpfen (s. Abbildung 4.14). Die einzelnen Detektoren sind prinzipiell als eine Black-Box zu sehen. Sie haben i Eingänge, welche sie mit anderen Detektoren oder den Aktuatoren verbinden. Die gemeinsame Funktion aller ist die Bereitstellung eines über die Zeit parametrisierten Ausgabewertes, welcher wiederum als Eingabebasis für andere Module dienen kann oder auf höchster Stufe direkt das Vorliegen einer bestimmten Geste repräsentiert. Auf Notationsebene entspricht eine Verschaltung mehrerer Detektoren der verschachtelten Benutzung von Prädikaten in den Prämissen anderer Prädikate. Soll beispielsweise ein Zugreifen eines fiktiven Objekts (z.Bsp. eines Rohres) mit beiden Hände erkannt werden, dann müssen beide Handposturen sowie die notwendigen Symmetriebedingung erfüllt sein.

Symmetrieerkennung

Um Merkmal 6, **interne Symmetrie** zu erkennen, wurde bereits ein Symmetrieaktuator konzipiert, um so Winkel und deren Veränderung zwischen den Symmetriestrahlen zu ermitteln (s. Abschnitt 4.2.4). Dieser ermöglicht in sehr knapper Form, unterschiedliche Symmetrien zwischen den Handreferenzsystemen auszudrücken. Eine Aussage, ob die Handflächen parallel zueinander stehen und in die gleiche Richtung weisen ist gegeben durch den Winkel zwischen den Sekundärstrahlen (`AngSekSek`):

$$\text{HOLDS?}(\texttt{IsPalmDirectionEqual}, i) := \text{HOLDS?}((\texttt{AngSekSek} < 10), i_1) \tag{4.62}$$

Folgende Definition notiert eine entgegengesetzte Ausrichtung der Daumen zueinander (indem der Winkel der Tertiärstrahlen `AngTerTer` getestet wird):

$$\text{HOLDS?}(\texttt{IsThumDirectionOpposite}, i) := \text{HOLDS?}((\texttt{AngTerTer} > 170), i_1) \tag{4.63}$$

Um ein Zugreifen wie es oben beschrieben wurde zu erkennen, werden die unterschiedlichen Teiltemplates miteinander verknüpft:

$$\begin{aligned}\text{HOLDS?}(\texttt{GraspThumbToThumb}, i) = {} & \text{HOLDS?}(\texttt{IsPalmDirectionEqual}, i_1) \\ & \wedge \text{HOLDS?}(\texttt{IsThumDirectionOpposite}, i_1) \\ & \wedge \text{HOLDS?}(\texttt{Grasp}, i_1) \\ & \wedge \text{HOLDS?}(\texttt{GraspLeft}, i_1)^{16}\end{aligned} \tag{4.64}$$

In Abbildung 4.14 wird eine hierarchische Zusammenschaltung mehrerer Einzelauswerter zur Erkennung der simultanen synchronisierten Greifgeste veranschaulicht. Auf unterster Ebene finden sich die sensorischen Eingangsdaten. Darüber liegen die Aktuatoren und darauf setzt eine einfache Hierarchie von Detektoren auf.

Eine solche hierarchische Verschaltung verhindert durch den konzeptuellen Aufbau Redundanzen durch Mehrfachberechnungen eines Attributs, welche auf Grund der Benutzung gleicher Attributsequenzen durch verschiedene Top-Level Detektoren hervorgerufen werden könnten. So wird in einer solchen Konzeption beispielsweise pro Hand nur ein Attribut zur Auswertung einer Greifpostur bereitgestellt. Verschiedene Templates können darauf aufsetzen. Ein Satz von Basisaktuatoren und Detektoren dient gleichzeitig der Detektion sehr verschiedener Gesten.

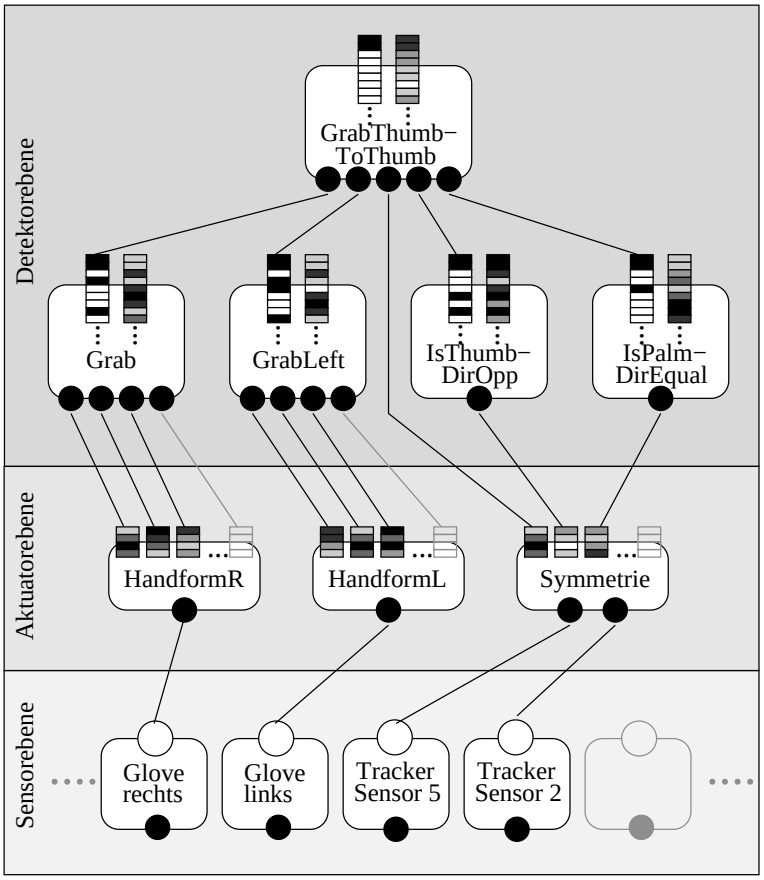

Abbildung 4.14: Ein Beispiel für eine hierarchische Detektorauswertung nach Gleichung (4.64): Die Auswertefunktion eines Detektors operiert nicht allein auf Aktuatorwerten, sondern greift auf die Attributsequenzen anderer Detektoren zu.

4.4 Motion-Modifikatoren

Für eine schritthaltende Auswertung kinemimischer/mimetischer Gesten, also der Interpretation der *grundlegenden* Bewegung — z.Bsp. im Hinblick auf die beschriebene Trajektorie — und die damit verbundene interaktive Veränderung eines Objektattributs (etwa der Lage oder der Position) in Echtzeit, müssen die Bewegungsinformationen einer Zwischenverarbeitung unterworfen werden. Zum einen beschreiben Benutzer durch Einschränkungen im menschlichen Bewegungsapparat keine *idealen* Trajektorien bei dem Bestreben geometrische Grundprimitive (etwa Kreise oder Richtungen/Geraden) gestisch anzudeuten (s. dazu auch [28], [69] und Abbildung 4.15). Weiterhin sollen auch zirkuläre Bewegungen, also Bewegungen, welche während der Ausführung ihre Anfangspunkte mehrfach durchschreiten können, durch ihre Wiederholung eine schritthaltende gestische Interaktion ermöglichen. Und schließlich müssen Möglichkeiten vorgesehen werden, um die angezeigten Veränderungen zu beschränken. So werden nur Veränderungen in einem vorher definierten festen Raster zugelassen, ähnlich dem Raster bei 2D- mausbasierten Zeichenprogrammen.

Diese Form einer toleranten Abbildung von unpräzisen Bewegungsinformationen auf präzise Objektveränderungen erleichtert einerseits die Manipulation und stellt andererseits auch eine Möglichkeit zur Einbindung weiterer Constraints zur Verfügung: Im CVK unterliegen beispielsweise aggregierte Objekte durch Freiheitsmatrizen modellierten verbindungsspezifischen Bewegungsbeschränkungen. Für eine Manipulation stehen so explizit nur noch bestimmte Freiheitsgrade zur Verfügung. Diese Einschränkung muss bei einer Interpretation kinemimischer/mimetischer Informationsvermittlung und deren interaktiver Umsetzung in gewünschte Veränderungen berücksichtigt werden.

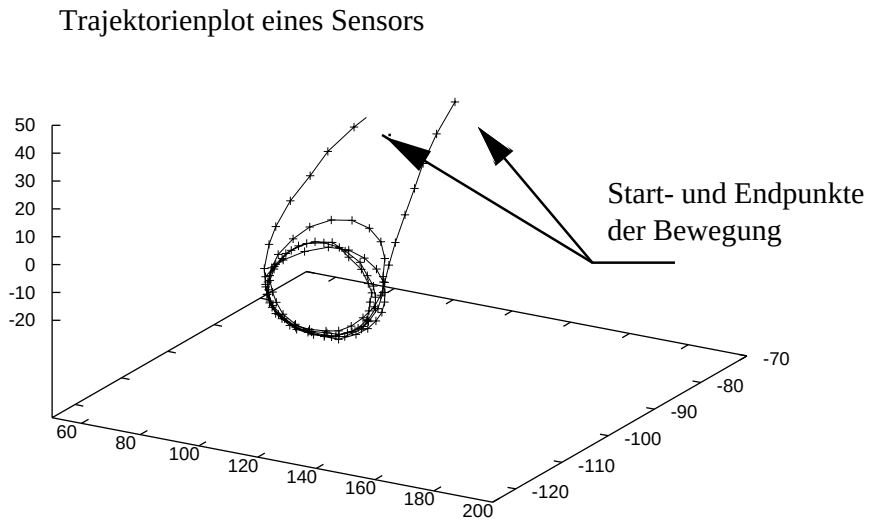

Abbildung 4.15: Exemplarischer Plot (gespiegelt) einer kreisförmigen Bewegung eines Arms auf Basis gemessener Sensordaten (n. [50]). Es ist eine deutliche Abweichung von einer angenommenen "idealen" Kreisbeschreibung zu erkennen.

4.4.1 Constraintmatrix

Vor einer Ausarbeitung der in diesem Abschnitt zu entwickelnden Funktionsträger wird einführend der Begriff der zu verändernden *Parameter* geklärt sowie eine formale Definition für die Spezifikation etwaiger Parameterbeschränkungen gegeben. Ein Parameter eines Objekts bildet ein Objektattribut auf einen nummerischen Wert ab. So kann beispielsweise die Lautstärke eines Geräusches durch einen einzigen Parameter (v), die Objektfarbe durch das Parametertripel (r,g,b) oder die Position durch (x,y,z) beschrieben werden. Die einzelnen Parameterwerte können dabei in vielen Fällen nur einen begrenzten Wertebereich einnehmen. Um diesen Wertebereich für unterschiedliche Parameter im Vorfeld anzugeben, wird das Konzept der **Constraintmatrix** eingeführt. Diese stellt im Prinzip eine allgemeine Form der in [46] entwickelten erweiterten Freiheits-Matrix dar (s. auch Abschnitt 2.3.1), da sie nicht explizit nur Positions- und Ausrichtungsparameter berücksichtigt, sondern die Parameterart wahlfrei hält.

Definition 11 (Constraintmatrix) *Eine Constraintmatrix C ist ein Container für relative Parameterbeschränkungen.*

$$\mathcal{C} := \begin{pmatrix} p_{11} & p_{12} \\ \vdots & \vdots \\ p_{m1} & p_{m2} \end{pmatrix} m \in \mathbb{N}, m \geq 1, \quad p_{ij} \in \mathbb{R}^+ \cup \{FREE,\ BLOCKED\} \tag{4.65}$$

Eine Constraintmatrix beinhaltet mindestens eine Zeile (einen Parameterfreiheitsgrad). Wie in [46] ausgeführt, geben die j Einträge den Wert des entsprechenden Freiheitssinn[17] wieder. Die Einträge $FREE$ und $BLOCKED$ bedeuten keine Einschränkung und vollständige Einschränkung in dem entsprechenden Parameterfreiheitssinn respektive. Die p_{ij} stellen für einen Parameter den Wert der möglichen Parameteränderung dar. Zwei Beispiele sollen dieses noch einmal verdeutlichen:

$$\mathcal{C}_{RotationX} := \begin{pmatrix} FREE & FREE \\ BLOCKED & BLOCKED \\ BLOCKED & BLOCKED \end{pmatrix} \tag{4.66}$$

$$\mathcal{C}_{Color} := \begin{pmatrix} 156.0 & 100.0 \\ 120.0 & 136.0 \\ 112.0 & 144.0 \\ 0.5 & 0.5 \end{pmatrix} \tag{4.67}$$

Die Rotations-Constraintmatrix (4.66) beschreibt (von oben nach unten) eine mögliche Rotation allein in der x-Achse, y und z sind blockiert. Die Farb-Constraintmatrix (4.67) repräsentiert die möglichen RGB- und α-Parameter mitsamt der noch möglichen Parameteränderungen durch die Angabe relativer Grenzen.

[17] Jeder Parameterfreiheitsgrad hat zwei Freiheitssinne. Diese repräsentieren die Veränderung in positiver oder in negativer Richtung.

Seien $p_{i1}, p_{i2} \in C_X$ zwei Einträge in einer Constraintmatrix. Sei weiterhin $\vec{r}$ eine potentielle Änderungsrichtungsangabe für den durch p_{i1} und p_{i2} repräsentierten Parameter. Dann gilt $FREE?(\vec{r}, C_X) = true$ genau dann, wenn die durch $\vec{r}$ beschriebene Änderungsrichtung nicht blockiert ist.

Durch das Konzept der Constraintmatrix lässt sich z.Bsp. das — im Hinblick auf die hier benötigten Informationen — speziellere Modell der erweiterten Freiheits-Matrizen für die gestische Parameterveränderung konvertieren: Je nach gewünschter Interaktion (z.Bsp. Rotation oder Translation) wird nur ein Teil der entsprechenden Freiheits-Matrix benötigt und kann direkt übernommen werden. So werden anwendungsspezifische Informationen, im Fall eines Einsatzes in der Domäne des CVK die Objektverbindungsarten, auf eine einheitliche Weise als abstrakte Parameterconstraints eingebunden.

4.4.2 Definition der Motion-Modifikatoren

Um die gestische Information einer interaktiv interpretierten Manipulation zugänglich zu machen, wird eine weitere neue Komponentenart in den AS-Graphen eingeführt, sogenannte **Motion-Modifikatoren**. Diese binden temporär — für die Dauer der Interaktion — an die bereits beschriebenen Attributsequenzen der Aktuatoren oder der Detektoren und dienen als Vermittler zwischen "unpräzisen" menschlichen Bewegungen und den möglichen Attributveränderungen eines zu manipulierenden Objekts.

In der folgenden Definition wird aus Gründen der Übersichtlichkeit auf die eigenständige Angabe der Eingabe- oder Quellattribusequenzen A_i und A_k als Tupelelemente verzichtet. Diese sind implizit durch die Funktion μ und der mit optionalem Parameter – gekennzeichnet durch $[A_k]$ – versehenen Funktion δ gegeben. Die von $\mathcal{R}$ unter Berücksichtigung der Constraintmatrix C freigegebenen möglichen Veränderungsparameter werden dabei als Rasterkandidaten bezeichnet.

Definition 12 (Motion-Modifikator) *Motion-Modifikatoren sind Bewegungsfilter und Kontrollinstanzen. Sie "mappen" die Bewegungsinformationen der Attributsequenzen in Simulationszeit-adäquaten Teilstücken und unter Berücksichtigung etwaiger Parameterbeschränkungen auf mögliche Attributparameter eines Szenenobjekts, bis ein Abbruchkriterium erfüllt ist. Ein Motion-Modifikator (kurz: MMod) ist ein 6-Tupel*

$$MMod = (\mathcal{R}, C, T_{eval}(t), \mathcal{H}, \mu(A_i, i), \delta_p(i, [A_k])) \tag{4.68}$$

Dabei sei

- $\mathcal{R} = \{\vec{r}_1, \ldots \vec{r}_t\}$, eine vorgegebene Menge von möglichen Vektoren — dem Raster — für die kontinuierliche Parameteränderung.
- C, eine Constraintmatrix mit eventuellen Einschränkungen der Parameteränderung.
- $T_{eval}(t) = \{t_k^e < t_{k+1}^e < \ldots < t_0^e \leq t, t_i^e \in \mathbb{N}_0, k \in \mathbb{Z}_0^-\}$, eine zeitabhängig wachsenden endliche Menge von diskreten, disjunkten und geordneten Simulationsschritten, den Traversierungszeitpunkten im Ablauf der Render-Loop.
- $\mathcal{H} = \{HOLDS?(< \texttt{Merkmal} >_1, i), \ldots\}$, eine nicht-leere Menge von Testprädikaten, den Abbruchkriterien für die Interaktion.
- $\mu(A_i, i)$, eine initiale Mappingfunktion von Werten einer Quellattributsequenz in einem Intervall i auf einen Parameterfreiheitssinn unter Berücksichtigung der von $\mathcal{R}$ und C freigegebenen Rasterkandidaten: $\mu(A_i, i) = \vec{r}_{ref}$, mit $\vec{r}_{ref} \in \mathcal{R} \wedge FREE?(\vec{r}_{ref}, C)$.
- $\delta_p(i, [A_k])$, eine kontinuierliche Mappingfunktion zur schritthaltenden Festlegung der relativen Parameteränderung in Abhängigkeit einer verstrichenen Zeit und einer optionalen Referenz-Quellattributsequenz.

Die Funktion eines Motion-Modifikators wird durch die explizite Einbindung in den Traversierungsablauf ausgelöst oder aktiviert. Diese Operation wird

als **binden** (bind) bezeichnet, da mit ihr eine Bindung mit Aktuatoren (ggfs. durch Sequenzattribute der Detektoren vermittelt) etabliert wird. Eine algorithmische Darstellung dieses Vorgangs ist folgendermaßen gegeben:

```
begin proc bind()
    set r_ref := μ(A_i, i);                ;;Parameterfreiheitssinn mappen
    for every newframe do                  ;;Framesynchrone Ausführung ⇒ T_eval(t)
        for every H? ∈ ℋ do                ;;Jedes Prädikat testen
            if H?(<Merkmal>, [t^e_{-1}, t^e_0]) = false then do
                unbind();                  ;;Bindung lösen
            done
        done
        δ_p([t^e_{-1}, t^e_0], [A_k]);     ;;relative Parameteränderung bestimmen
    done
end proc
```

Abbildung 4.16: Der Aktivierungsalgorithmus eines Motion-Modifikators verdeutlicht den Aufbau einer Bindung und das darauf folgende kontinuierliche Auswerten der Mappingfunktion für jeden Simulationsschritt.

4.4.3 Ein Rotations-Motion-Modifikator

An einem Beispiel soll die Funktionsweise eines Motion-Modifikators näher erläutert werden. Dieser bildet kreisförmige Bewegungen einer Hand auf eine Änderung der Ausrichtung eines nicht verbundenen Objekts ab. Das Objekt ist frei im Raum platziert und unterliegt keiner weiteren Bewegungseinschränkung. Dafür werden die benötigten Komponenten wie folgt definiert. Zuerst wird das Bewegungsraster festgelegt, dieses geschieht zur Veranschaulichung nur unter Benutzung der drei Hauptachsrichtungen des Basissystems:

$$\mathcal{R} = \left\{ \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}-1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\-1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}, \begin{pmatrix}0\\0\\-1\end{pmatrix} \right\} \tag{4.69}$$

Die Constraintmatrix für eine nicht beschränkte Rotation lautet:

$$\mathcal{C} = \mathcal{C}_{Rotation} := \begin{pmatrix} FREE & FREE \\ FREE & FREE \\ FREE & FREE \end{pmatrix} \tag{4.70}$$

$T_{eval}(t)$ ist — wie bereits für andere AS-Graph Komponenten verdeutlicht wurde — durch die Traversierung während der Simulation festgelegt. Die Abbruchkriterien sind für diese Form der Bewegung bereits in Abschnitt 4.3 (S. 113) durch Gleichung (4.58) in einem einzigen Testkriterium zusammengefasst worden:

$$\mathcal{H} = \{\text{HOLDS?}(\texttt{Orbitting}, i)\} \tag{4.71}$$

Die initiale Mappingfunktion μ sucht zu Beginn der Interaktion den geeigneten Rasterkandidaten — im Beispiel die benötigte Rotationsachse — aus $\mathcal{R}$. Durch die Art der direkten Abbildung eines spatialen Bewegungsparameters auf einen adäquaten Manipulationsparameter bietet sich hier ein einfacher Winkelvergleich als Testfunktion an, ein *direktes spatiales mapping*. Die benötigte Vergleichsnormale der Bewegung findet sich nach Gleichung (4.34) in einer Attributsequenz eines Aktuators (s. Abschnitt 4.2.4, S. 98):

$$\begin{gathered}\mu((\texttt{Norm}, \vec{a}_n(t')), i) = r_x \text{ mit} \\ r_x \in \mathcal{R}, \vec{t} = \varnothing(\{\vec{a}_n(t') \mid t' \in i\}) : \nexists\, r_y \in \mathcal{R} : r_y \cdot \vec{t} < r_x \cdot \vec{t}\end{gathered} \tag{4.72}$$

Der relative Änderungswert des Stellparameters wird durch $\delta_p(i, [A_k])$ bestimmt. A_k ist dabei als optional in dem Sinne anzusehen, dass es auch möglich ist eine konstante Parameteränderung zu verwenden. Im Beispiel wird von dieser Möglichkeit dagegen kein Gebrauch gemacht. Die Geschwindigkeit der ausgeführten Bewegung dient hier der Festlegung der relativen Änderung nach Gleichung (4.23) (ebenfalls n. Abschnitt 4.2.4, S. 98). Sei $i = [t_a, t_e]$, dann gelte:

$$\delta_p(i, (\texttt{AvgSpeed}, \vec{a}_s(t')) = \begin{cases} 90^\circ * (t_e - t_a) & \text{wenn } \varnothing(\{\vec{a}_s(t') \mid t' \in i\}) < 0.5 \\ 360^\circ * (t_e - t_a) & \text{wenn } \varnothing(\{\vec{a}_s(t') \mid t' \in i\}) > 1.0 \\ 180^\circ * (t_e - t_a) & \text{sonst.} \end{cases} \tag{4.73}$$

Der zeitabhängige Wert der Parameterveränderung richtet sich nach der Bewegungsgeschwindigkeit und verläuft in einem dreistufigen Raster. Die Drehrichtung ist durch die Art der Berechnung der Attributsequenz $(\texttt{Norm}, \vec{a}_n(t'))$ durch den Aktuator und die Festlegung der möglichen antiparallel vordefinierten Drehachsen bestimmt. Sie verläuft bei Sicht in Richtung der Drehachse im Uhrzeigersinn.

Durch den so gegebenen Rotations-Motion-Modifikator kann die relative Orientierungsänderung eines vorher zu spezifizierenden Objekts mittels

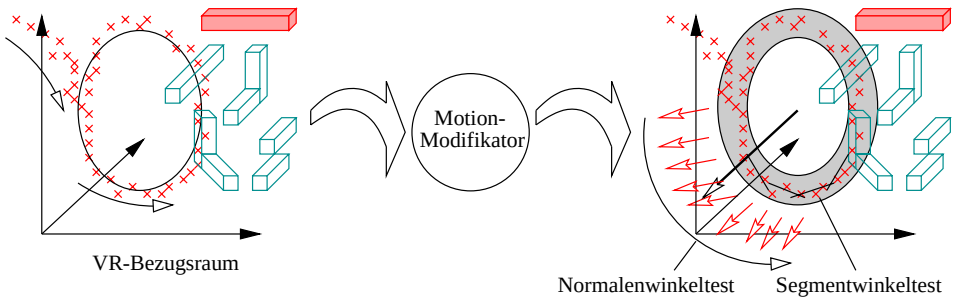

Abbildung 4.17: Ein Motion-Modifikator bei der beispielhaften Abstraktion einer ausgeführten Rotationsbewegung über initiale Bestimmung der Rotationsachse (dem Freiheitssinn) und anschließender Evaluation von Testprädikaten auf Basis von Normalen- und Winkelvergleichen.

einer kontinuierlichen mimetisch/kinemimischen Interaktion für jeden Simulationsschritt bestimmt werden. Das Abbruchkriterium überprüft die verschiedenen Bahncharakteristiken der beschriebenen Bewegung. In Abbildung 4.17 wird der Schritt der Bewegungsabstraktion, die Festlegung eines initialen Freiheitssinnes und die daraufhin u.a. berücksichtigten Winkelvergleiche dargestellt.

Ein Motion-Modifikator arbeitet unabhängig von einem gegebenen Basissystem. Wenn Transformationen benötigt werden, um Werte aus den Bewegungsattributsequenzen mit Werten aus $\mathcal{C}$ und $\mathcal{R}$ in Bezug zu setzen, so sind diese Transformationen bei der Festlegung von μ zu berücksichtigen. Die im CVK verwendeten erweiterten Freiheits-Matrizen sind z.Bsp. relativ im jeweiligen Port-Koordinatensystem definiert. Etwaige Rasterwerte werden für diesen Fall ebenfalls relativ zu dieser Basis festgelegt.

Die Weitergabe der relativen Parameterwerte erfolgt durch eine Bindung auf der Ausgabeseite des Motion-Modifikators. Abbildung 4.18 zeigt die Weitergabe sowohl des initialen Wertes der Drehachse als auch der kontinuierlichen Drehwinkelwerte für die Rotationsvermittlung. Empfänger dieser Informationen sind spezialisierte Module zur Manipulation von Objekten in der Szene, so genannte *Manipulatoren* (s.n. Abschnitt). Die Motion-Modifikatoren sind die speziellen Bindeglieder zwischen den von Anwendern geäußerten Bewegungen und einer möglichen Interpretation im Hinblick auf eine kontinuierliche Parameteränderung.

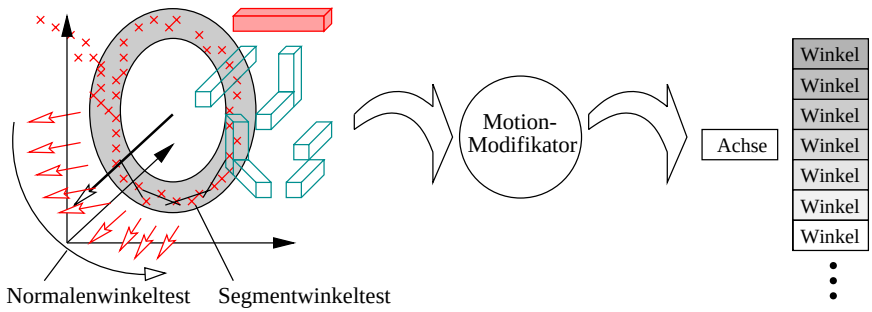

Abbildung 4.18: Ermittlung Simulationsschritt-diskreter Teilmanipulationen (hier Rotationen) gegeben durch die Mappingfunktionen eines Motion-Modifikators.

4.4.4 Manipulatoren

Die Weitergabe der diskretisierten Manipulationsschritte geschieht über die Bindungen der Motion-Modifikatoren an Instanzen eines weiteren Komponententypus: Die verschiedenen Parameteränderungswerte werden von — für die Art der Parameter — dedizierten **Manipulatoren** entgegengenommen. Ein sogenanntes Manipulatorenkonzept findet sich auch in vielen mausbasierten Schnittstellen zu 3D-Anwendungen (s. z.Bsp. in [99]), dort handelt es sich um einen einzigen integralen und monolithischen Bestandteil zur Auswertung von Benutzereingaben (hauptsächlich Mausbewegungen) und zur anschließenden Manipulationen eines Zielobjekts. Manipulatoren sind Szenengraphobjekte, sie ersetzen im Normalfall andere Szenengraphknoten durch eine editierbare Version dieser. Durch die Auslösung bestimmter Events werden die Parameter des Knotens daraufhin verändert.

In der hier entworfenen Konzeption wird dieses Vorgehen modularisiert, um die Grundlagen für die vorgeschaltete gestische Analyse zu schaffen. Manipulatoren erhalten dabei die bereits für sie vorgesehene Aufgabenspezifikation. Sie sind interaktionsspezifisch, z.Bsp. speziell für Rotationen, Translationen oder Farbänderungen ausgelegt, und erhalten während einer mimetischen/kinemimischen Interaktion die benötigten Parameter als Sequenzen von auszuführenden Teilmanipulationen (s. Abbildung 4.19) von den Motion-Modifikatoren.

Diese Manipulationen werden daraufhin direkt auf das Zielobjekt angewendet. Auf Manipulatorenebene ist es vollständig opak, ob die Parameter wie bisher von einem Mausinterface geliefert werden oder ob sie das Resultat

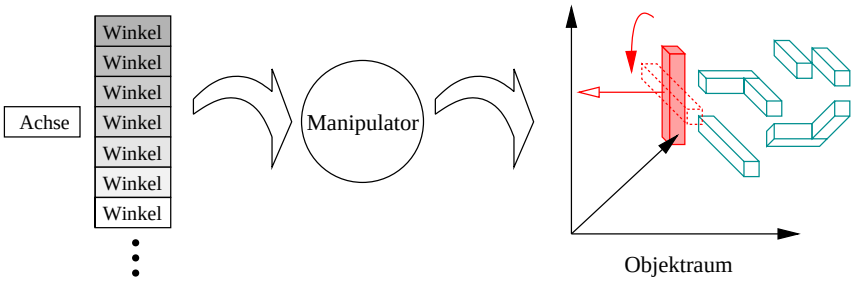

Abbildung 4.19: Weitergabe diskreter Teilmanipulationen an einen Manipulator zur Ausführung einer Rotation.

einer mitunter sehr vielschichtigen Zwischenverarbeitung von Aktuatoren, Detektoren und Motion-Modifikatoren sind. Insbesondere machen Manipulatoren keine Annahme über die Form der Parameterbereitstellung. Durch diesen Umstand lassen sich ebenfalls andere — nicht in erster Linie spatial zu interpretierende — Manipulationen durch Gesten vermitteln. Ein entsprechender Motion-Modifikator wird in diesem Fall anstatt eines direkten spatialen mappings (s. Abschnitt 4.4.3) eine Art *metaphorisches mapping* definieren. Die Verwendung unterschiedlicher mapping-Funktionen und spezieller Manipulatoren führt so durch die gleiche Gestenform anstatt eine Verschiebung ebenso etwa eine Farbänderung eines Objekts herbei. Der metaphorische Transfer zwischen Gestikraum und Farbraum wird durch das mapping etabliert.

Der ausgearbeitete Rotations-Motion-Modifikator und die in diesem Abschnitt angestellten Betrachtungen über die Art der mapping-Funktion verdeutlichen das konzeptionelle Vorgehen bei der Umsetzung vieler der in Abschnitt 2.3.2 identifizierten Interaktionen und Manipulationen in Virtuellen Umgebungen durch eine kontinuierliche Interpretation gestischer Eingaben.

4.5 Zusammenfassung

Mit den Konzepten Aktuator, Detektor, Motion-Modifikator und Manipulator wurden Basiskomponenten entwickelt, um gestische Eingaben unter den Randbedingungen einer VR-Anwendung als Interaktionsmedium nutzbar zu machen. Aktuatoren abstrahieren von der eigentlichen Sensoreingabe und stellen die für die Gestik wesentlichen Bewegungsmerkmale synchron und in einem einheitlichen Format zur Verfügung. Diese Informationen werden von

Detektoren entgegengenommen, analysiert und klassifiziert. Einzelne Detektoren führen dabei nur relativ einfache Berechnungs-, Bewertungs- oder Vergleichsoperationen durch, lassen sich aber als atomare Bausteine zu komplexen Erkennungsnetzwerken zusammensetzen, deren interne Berechnungsvorschriften durch die resultierenden Netzstrukturen definiert sind. Solche Netzwerke — und damit deren Funktion — können flexibel erweitert oder angepasst werden.

Die Umsetzung kontinuierlicher Interaktionen war Grundlage der Ausarbeitung der Motion-Modifikatoren. Sie kapseln die hier benötigten Kontroll- und Abbildungsfunktionen, letztere unter Berücksichtigung etwaiger Veränderungsconstraints eines Objekts. Manipulatoren wenden daraufhin die Parameteränderungen direkt an.

Mit den erarbeiteten Konzepten stehen im weiteren Verlauf grundlegende Komponenten zur Verfügung, um gestische Eingaben zu analysieren, zu detektieren und um die angestrebten Manipulationen durchzuführen. Durch die Zerlegung der Teilaufgaben in einzelne funktionale Module mit definierten Ein-/Ausgabeschnittstellen über die Attributsequenzen wird es ermöglicht, die Verarbeitung gestischer Eingaben in Form eines Baukastensystems umzusetzen und einer VR-Anwendung zur Verfügung zu stellen.

Kapitel 5

ATN-basierte Auswertung multimodaler Interaktionen

5.1 Analyse sprachlicher Handlungsanweisungen

Dieser Abschnitt befasst sich mit sprachlichen Äußerungen im Kontext der Interaktion mit einem VR-System — mit Handlungsanweisungen. An konkreten Beispielen aus der Domäne der VK werden exemplarische verbale Ausdrücke analysiert, indem sie im Hinblick auf die sie bildenden Konstituenten, die grammatikalische Struktur sowie den semantischen Gehalt, untersucht werden. Die semantische Analyse der Teiläußerungen führt daraufhin zur Identifikation von zusammengehörigen Konstituenten mit gleichem kommunikativem Bezug. Diese Gruppierungen der Einheiten bilden anschließend die Basis, um multimodale Eingaben zu interpretieren und auszuwerten. Sie sind damit die Mittlerebene zwischen Sprache, Gestik und Anwendung in Bezug auf den Sinngehalt der jeweiligen Äußerung. Der Vergleich mit den bereits analysierten Gestenformen und ihrem Kommunikationsgehalt ermöglicht die Identifikation ähnlicher, identischer oder sich ergänzender *gestischer* Information. Diese Querbezüge dienen als Grundlage für die hier entwickelte technische Integration der sprach-gestischen Eingaben.

5.1.1 Exemplarische verbale Äußerungen

Die folgenden Beispiele bilden eine Teilmenge der insgesamt im CVK möglichen sprachlichen Äußerungen, speziell auch solche, in denen eine gestische Expression auf Grund von verbalen Unterspezifizierungen oder Ambiguitäten zu erwarten ist (sprach-gestische Beispiele während der Disambiguie-

rung[1] sind zu finden in [94]). Es werden in diesem Kontext aber auch solche Sprachbeispiele untersucht, welche nach Abschnitt 2.3.2 nicht nur speziell in Anwendungen der VK Verwendung finden, sondern allgemein für VR-Umgebungen geeignet sind. Hier wird besonders die Möglichkeit der Selektion in den Vordergrund gestellt. Das Festlegen von referenzierten Objekten oder Orten ist eine wichtige Grundfunktionalität in graphischen Systemen, da so ein direktes Feedback der Auswirkung der nächsten Interaktion ermöglicht wird.

Navigation:

"Gehe zu Spot vier"
"Gehe da hin"
"Gehe nach links"

Aktivierung:

"Öffne Schott eins"
"Öffne das Fenster"
"Kalibriere die rechte Hand"

Erzeugung:

"Gib mir ein Rad"
"Erzeuge ein gelbes Rohr
"Gib mir eine grüne lange Stange"

Selektion:

"Nimm dieses Teil"
"Nimm das rote Teil
"Nimm dieses rote Rohr"
"Nimm das linke gelbe runde Teil"

Verbindung:

"Setz es da dran"
"Verbinde dieses mit dem"
"Setz dieses rote Rohr an das"

[1] Disambiguierung bezeichne hier den Prozess der schrittweisen Konkretisierung sprachlich und/oder gestisch übermittelter Referenzen und Handlungen.

"Verbinde dieses gelbe Rohr mit diesem blauen Teil"
"Ich möchte diese Heckverkleidung mit diesem roten Teil verbinden"

Trennung:

"Trenne die Teile"
"Trenne das rote Rohr"
"Trenne das blaue Rohr vom gelben Rohr"

Transformation:

"Dreh das Rad so"
"Dreh dieses Rohr um 30 Grad"
"Dreh dieses Rad um 45 Grad um diese Achse"
"Verschiebe dieses Teil nach da"
"Und nun dreh das Teil so"

Kombinationen:

"Nimm das rote Teil und setze es an das blaue lange Rohr"
"Nimm das gelbe Rad und dreh es so herum"
"Und jetzt nimm die Heckverkleidung und setze sie da dran"

Um einen Ausgangspunkt für eine Interpretation dieser sprachlichen Ausdrücke zu ermitteln, werden im Folgenden die strukturellen und semantischen Gemeinsamkeiten herausgestellt. An Beispielen lassen sich dabei die bereits abzusehenden Auswirkungen auf einen automatischen Analyse- und Interpretationsvorgang herausarbeiten.

5.1.2 Struktur und Semantik der Handlungsanweisungen

Die Formen der beschriebenen Handlungsanweisungen besitzen einen ähnlichen Aufbau. Als konstituierende Einheiten können grundsätzlich die Aktions- und Referenzbeschreibungen identifiziert werden. Gewünschte durchzuführende Manipulationen werden durch die Aktionsdeskriptoren festgelegt. Es sind die Prädikate — die Verben — in den Verbalphrasen (VPs). Sie treten in der Aktivform einer nicht konjugierten Handlungsanweisung in einfacher Form auf. Einfache Beispiele sind *"nimm"*, *"verbinde"* oder *"drehe"*. Durch diese Aktionsdeskriptoren wird eindeutig die Art der Interaktion

festgelegt. Grundlage dafür sind in einem technischen System die überhaupt zur Verfügung gestellten Manipulationsmöglichkeiten.

Die Beispiele beginnen in der Regel mit den Prädikaten, da in den Aktivform-Konstrukten häufig die folgende Transformationsregel zur Modifikation der Grundstruktur angewendet wird: Bei diesen imperativen Phrasen entfällt oft das Subjekt des Satzes, eine sogenannte Subjekt-Deleatur. Das Prädikat steht dadurch am Beginn des Satzes und leitet diesen ein. Dieses geschieht durch eine implizite Rollenvergabe: Das System nimmt die Stelle des Kommunikationspartners ein, somit muss im Gebrauch der Imperativsätze das Subjekt nicht mehr explizit gemacht werden. Dieser Umstand lässt sich aber nicht verallgemeinern. Wird kein Imperativ benutzt, so findet in einem unterschiedlichen Konstrukt die Passivform Anwendung. Der Anwender sieht die umzusetzende Manipulation als Wunsch an, welchen er als Interaktionsziel äußert. Grammatikalisch ist zu beachten, dass in einem solchen Fall das die Handlung ausdrückende Prädikat nicht der ersten auftretenden VP entspricht, sondern als zweite VP eventuell zusätzlich im Kontext einer Präpositionalphrase (PP) auftritt. Die noch folgenden Ausführungen beschränken sich aber o.B.d.A. auf die Betrachtung der Aktivform, da sie im Kontext der Interaktion die deutlich prägnantere Ausdrucksform erlaubt.

Aktionsbeschreibung

Das Prädikat ist die "treibende" Kraft der verbalen Ausdrücke. Je nach Typus werden für die verschiedenen Aktionen jeweils eine unterschiedliche Anzahl von Objekt- und/oder Ortsreferenzen benötigt. Durch das Verb der VP wird die gewünschte Interaktion konkret benannt und die zur *Ergänzung* benötigten weiteren Konstituenten bestimmt. Die Art des Prädikats hat also bestimmte **Subkategorisierungseigenschaften**. Durch diese werden zwar — im Sinne einer Verallgemeinerung — gewisse grammatikalische Gemeinsamkeiten der Verben aufgegeben, im Fall der hier im Kontext gebrauchten wenigen Verben ist dieses aber zu vernachlässigen. Das Verb *gehen* verlangt z.Bsp. nach einer Adverbialergänzung, um den Zielort oder die Zielrichtung zu beschreiben. Die einfachen Aktivierungsverben sind *transitiv*, die Erzeugung und die Selektion benötigen je eine zusätzliche Ergänzung, welche das oder die entsprechende(n) Objekt(e) beschreibt oder beschreiben. Ein Verbindungsverb benötigt ebenfalls eine Objektbeschreibung und daraufhin ein Präpositionalobjekt.

Objektbeschreibung

Neben der Information des Prädikats, welches die Art der Manipulation bekannt gibt, finden sich die Referenzen zu Objekten, welche für die spezifische Handlung benötigten werden, entweder in entsprechenden Pronomen (falls die Referenten aus dem Kontext schon bekannt sind) oder in einer oder mehreren *NPs* (Nominalphrasen). NPs unterscheiden sich grammatikalisch in *DNPs* (Definite Nominalphrasen) und *INPs* (Indefinite Nominalphrasen). Erstere sind Konstrukte, welche eine spezielle Instanz oder spezielle Instanzen eines Objekts oder einer Objektklasse im Diskurs referenzieren. Beispiele: " ... *die gelbe linke Leiste* ... " meint genau die eine Leiste, " ... *die gelben linken Leisten* ... " meint dagegen alle gelben Leisten, welche links liegen. Die INPs werden dagegen nicht durch einen definiten, sondern durch einen indefiniten Artikel eingeleitet und nehmen nicht direkten Bezug auf eine konkrete Instanz:" ... *ein blaues Rohr* ... ". NPs bestehen im Prinzip aus Artikeln, einer eventuell leeren Menge von Eigenschaftsbezeichnern — Adjektiven und Lokaladverbien — und einem entsprechenden Typbezeichner. Ausschließende absolute Attribute, etwa **Farb**- oder **Form**bezeichnungen, schränken als Bestimmungskriterien die möglichen in Frage kommenden Objekte in eindeutiger Weise ein. **Lage**- und **Orientierungs**attribute — wie in *"das linke Rohr"* — sind relative Beschreibungen und dienen der weiteren Disambiguierung der nach den absoluten Kriterien "ausgefilterten" Objekte. Hier finden wir die Analogie zu den in Abschnitt 2.3.2 beschriebenen relativen Lage- und Erscheinungseigenschaften. Um die Objekte näher zu spezifizieren, werden sie nach ihren "ersichtlichen" Attributen benannt. So lassen sich Mehrdeutigkeiten bei ähnlichen oder gleichen Objekten in begrenztem Maße sehr effektiv auflösen.

Für allgemeine Objektbeschreibungen ist dieses stark vereinfacht. Durch den rekursiven Sprachaufbau ist es uns aber ohne weiteres möglich, mit Hilfe komplexer Relativsatzkonstrukte oder aber durch eingeschachtelte PPs Objekte immer feiner zu beschreiben. Die folgenden Ausführungen beschränken sich hier auf den NP-Fall ohne Relativkonstrukt.

Modal- und Lokalhinweise

Sowohl für die Aktions- als auch für die Objektbeschreibung wurden bisher nur die einfachen oder — in Bezug auf die Subkategorisierung der Verben — zwingenden Ergänzungen eingegangen. Eine im Kontext multimodaler Eingaben für 3D-Szenen sehr bedeutsame Art von (Teil-)Äußerungen, sind besonders **Modaladverbien** (*so, wie* ...), **Lokaladverbien** (*dahin, hier,*

da ...) oder PPs, welche modalen oder lokalen Bezug explizit machen (... *auf diese Art,* ... *von links oben*).

Diese Konstituenten und Konstrukte bekommen besonderes Gewicht bei der Interpretation sprach-gestischer Äußerungen, da sie eindeutig Bezug auf Informationen nehmen, welche durch Gestik vermittelt werden können. Neben den unbedingt benötigten Objektreferenzen eines Prädikats wird beispielsweise die mit dem Prädikat beschriebene Handlung durch weitere Ergänzungen mit Teilen der hier beschriebenen Art weiter konkretisiert (... *dreh das Teil so* ...). Auch bei detaillierterer Objektbeschreibung über einfache NPs hinaus finden sich die Modal- und Lokalhinweise (... *das Rad, welches so auf dem Rohr steckt.*).

5.2 Sprachgestützte Gestik - Multimodalität

Ein Anwender kommt bei einer rein verbalen Informationsvermittlung räumlicher Beziehungen, von Formen und Bewegungen schnell in Situationen, in welchen die sprachlichen Ausdrucksmöglichkeiten entweder zu abstrakt sind oder unser internes Bild über das zu kommunizierende Faktum zu unpräzise ist, um Attribute direkt, über die Angabe konkreter Werte, zu quantisieren. Daneben kommt es besonders bei der Benutzung von vielen formgleichen Objekten — wie den parametrisierten baukastenbasierten Konstruktionsteilen im CVK — zu Mehrdeutigkeiten, da sprachliche Objekteigenschaftsbeschreibungen bei vielen ähnlichen oder gleichen Attributen Objekte nicht ausreichend disambiguieren können. Räumliche Beziehungen und Eigenschaften wurden in Abschnitt 2.3.2 aber als primäre Interaktionsziele eines Eingabesystems für die VK identifiziert. Betrachten wir Ortsattribute (linke, rechte, über, neben ...), so haben diese relativen Bezug und geben nur qualitativ eine beschriebene Lage wieder. Auch Größen- und Formangaben können so spezifiziert werden. Für die Vermittlung der Ausprägung dieser Attribute ist Sprache aber nur dann als ausreichend anzusehen, wenn der Wert der Veränderung durch konkrete Zahlenwerte spezifiziert werden kann — beispielsweise, wenn bei einer Drehung die Achse bekannt ist und um genau 30 Grad im Uhrzeigersinn gedreht werden soll.

Interaktive Systemen sind dagegen hochgradig explorativ. Gewünschte Manipulationen können selten konkret und vollständig angegeben werden, sondern werden in der laufenden Interaktion moduliert. Gerade diese Eigenschaft stellt eine der Stärken dieser Systeme dar. Betrachten wir die gewünschten Interaktionen in der VR und speziell im CVK, können sprachliche Unzulänglichkeiten durch multimodale Eingaben ausgeglichen werden.

5.2.1 Semantische Zusammenhänge

Nach den Betrachtungen des kommunikativen Gehalts sowohl der Gestik als auch der verbalen Handlungsanweisungen, werden nun die semantischen Querbezüge zwischen beiden Modalitäten ermittelt. Die Basis der technischen Integration beider Modalitäten sind dabei die in Abschnitt 2.2.3 erörterten Ergebnisse von [12],[42] und [65]: eine temporale Kopplung zwischen *bestimmten* sprachlichen und gestischen Merkmalen bestätigt experimentell einen semantischen Zusammenhang der mit den Modalitäten vermittelten Information. An dieser Stelle wird nicht auf die Art dieser Merkmale eingegangen, sondern auf die Identifikation der semantischen Einheiten für eine spätere Integration. *Gleichwertige* Informationseinheiten finden sich im verbalen Kanal sowohl auf komplexen zusammengesetzten Strukturen, einzelnen Phrasen, als auch in der Semantik einzelner Konstituenten.

Deixis und Blickrichtung

Die Objekt- und Ortsreferenzen kommen in allen bisher besprochenen Interaktionsarten mindestens in einer Instanz vor. Sie sind syntaktisch in den Nominalphrasen der verbalen Ausdrücke zu finden. Wie wir in Abschnitt 2.4.2 gesehen haben, spielt Benutzerdeixis bei der Vermittlung räumlicher Beziehungen mittels extern gerichteter Bedeutungen in Bezug auf die eigene körperliche Präsenz eine primäre Rolle. Verbindet man diese mit verbalen Äußerungen zu Objekt- oder Ortsreferenzen, so erhält man komplementäre multimodale Referenzbeschreibungen. Die Blickrichtung bestimmt während einer Äußerung die relative Sicht, und damit das Ankerreferenzsystem für relative Lagebeschreibungen, aber auch den globalen Sichtbereich, also resultierend die Objekte, welche überhaupt im Betrachtungsfokus liegen. Die Semantik von **Lokalattributen** ist daher durch die Benutzersicht bestimmt.

Zeigegesten

Zeigegesten werden zur Konkretisierung eines Ortes, einer Richtung oder zur Bestimmung von Objekten in dieser Richtung eingesetzt. Die Äußerung referenzbeschreibender Attribute erfolgt relativ zu dem aktuellen Benutzerreferenzsystem und absolut auf der durch Blick-/Zeigerichtung fokussierten Untermenge von Objekten. Eine temporale Kopplung einer Zeigegestenklimax ist an allen Stellen einer DNP-Äußerung möglich. Besonders dominant ist hier aber eine Kopplung (im Sinne eines kurzen zeitlichen Vorlaufs der Geste vor dem entsprechendenWort) bei **Demonstrativpronomen**, **Artikeln**, **Lokalattributen** und **Lokaladverbien**.

Mimetisch/Kinemimische und ikonische Gesten

Die Manipulationsarten können durch die verbalen Aktionsdeskriptoren eingeleitet werden. Auf das Prädikat folgende Präpositionalphrasen oder Adverbialkonstrukte konkretisieren eine gewünschte Manipulation und liefern ggfs. weitere benötigte Manipulationsinformationen. Im gestischen Kanal dienen hier mimetisch/kinemimische Gesten dem "Vormachen" oder der Modulation der gewünschten Interaktion. Angezeigt wird dieses durch den Gebrauch von **Modaladverbien** und durch den Typus des Prädikats selber, wenn beispielsweise „ ... *schieb* ... " geäußert wird und gleichzeitige eine expressive Translationsbewegung einer Extremität erfolgt.

Adverbialkonstrukte und Präpositionalphrasen dienen im gewählten Szenario nicht allein der Konkretisierung einer vermittelten Manipulation, sondern sie können auch zur Disambiguierung von Objekten eingesetzt werden. Werden in einem solchen Fall gestische Informationen vermittelt, so werden diese Gesten — gemäß der Funktion — als ikonisch bezeichnet. Hier wird einmal mehr die nahe Verwandschaft von mimetischen und ikonischen Gesten offensichtlich. Beide drücken spatiale Primitive aus, einmal werden diese als Aktion, ein anderes Mal als statische (Objekt-)Eigenschaft interpretiert. Diese Eigenschaften werden ebenfalls durch **Formattributadjektive** (wie *lang, dünn, rund* ...) kommuniziert. Wird die Geste während einer zu einer Objektbeschreibung gehörenden Phrase geäußert, so hat sie spatiographischen Charakter. Gehört die Phrase zur VP, so ist es eine pictomimische Geste, dies erschwert die Unterscheidung zu mimetischen/kinemimischen Gesten.

5.3 Ein erweitertes ATN für die multimodale Integration

5.3.1 Voraussetzungen

Die zuvor phänomenologisch geschilderten Zusammenhänge zwischen Gestik und Sprache bilden die Grundlage für das technische Integrationssystem. Konzeptuell ist sowohl eine grammatische und eine semantische Analyse der Sprache als auch eine Klassifikation und weitestgehende Interpretation der Gestik notwendig, um die zeitlichen und die inhaltlichen Bezüge herzustellen. Wie in Abschnitt 5.2 herausgearbeitet wurde, weisen einerseits einzelne bestimmte Wortarten auf gestische Artikulation hin. Andererseits stellt der Kontext der aktuellen Äußerung — häufig in Phrasenstrukturen zusammenzufassen — den Sinnbezug her. Umgekehrt weisen aber ebenso bestimmte Gestenformen auf den inhaltlichen Bezug hin. Eine Geste stellt wie die be-

sprochenen sprachlichen Konstrukte einen inhaltlichen Kontext her: Eine Zeigegeste deutet z.Bsp. auch ohne die Betrachtung des verbalen Kanals auf Orte oder Objekte in der bezeigten Richtung hin. Beide Kanäle dienen dazu, *mentale* Konzepte zu kommunizieren. Nach McNeill [67] diversifizieren sich diese Konzepte am growth point in die einzelnen Modalitäten und begründen damit den Zusammenhang zwischen dem temporalen und dem semantisch-pragmatischem Synchronismus (s. dazu Abschnitt 2.2.3). Es wird also eine Methode benötigt, um sowohl inhaltliche Bezüge über den zeitlichen Zusammenhang als auch über den Kommunikationskontext und die Äußerungssemantik herzustellen.

Die hier betrachteten Gestenformen stellen dabei eine weitere Anforderung durch ihre zeitliche Äußerungsdauer. Eine Zeigegeste kann als atomares abgeschlossenes Event aufgefasst werden, wenn man beispielsweise *Aktion und Pause* der entsprechenden Hand betrachtet (s. Abschnitt 4.3.2). Eine mimetisch/kinemimische Geste aber hat i.A. eine ausgedehnte zeitliche Dauer und ist zu einem beliebigen Zeitpunkt nicht zwangsläufig als abgeschlossen zu erkennen. Die hier benötigte Gestenklassifikationsfunktionen steht durch die Attributsequenzen bereits zur Verfügung, welche sowohl mit einem Event- als auch mit einem Intervallmodell kompatibel sind. Ein entsprechender multimodaler Integrationsformalismus muss daher sowohl diskrete Events — z.Bsp. den Beginn einer Zeigegeste — als auch zum Integrationszeitpunkt evtl. noch andauernde Klassifikationsergebnisse mit ausgedehnter Zeitlichkeit — etwa die noch ausgeführte Drehung einer Hand — berücksichtigen.

5.3.2 Methodenübersicht

Um die im letzten Abschnitt beschriebenen multimodalen Zusammenhänge zu modellieren und daraufhin technisch umzusetzen, muss eine geeignete Methode Verwendung finden. Dabei wurden in der Vergangenheit unterschiedliche Ansätze verfolgt. Einer der ersten war die restriktive Festlegung des Zeitpunkts, wann genau eine Geste in einer sprachlichen Äußerung vorkommen durfte (bei Bolt [8]), ohne dieses formal explizit zu machen. Ebenso verwenden auch Neal und Shapiro [1], Koons et al [45], Lenzman [55] und andere eine sprachgetriebene Integration. Die Sprachanalyse liefert hier z.Bsp. *Gestenplätze* [55] in einer Type-Token Repräsentation der Sprachinformationen, welche als mögliche Integrationsziele (bei [55] bezogen auf Zeigegesten) dienen.

Andere Ansätze fassen solche Type-Token Strukturen zusammen und benutzen als Integrationseinheiten — vor allem häufig bei Stift-basierten multimodalen Eingabesystemen — eine Framenotation der jeweils gesammelten multimodalen Information (z.Bsp. bei Koons et al [45], Vo und Wood [93]).

Das Wissen über die Integrationsfunktion liegt in allen diesen Ansätzen in prozeduraler Form vor: Was womit gematcht (abgeglichen) werden darf, hängt allein von diesen Regeln ab.

Johnston [36] (Johnston et al [37]) führen die Frame-basierten Integrationsmethoden weiter. Die Informationen werden hier in sogenannten *feature structures* zusammengefasst. Diese Attributmatrizen können rekursiv aufgebaut werden, da ein entsprechender Attributwert wiederum eine feature structure beinhalten kann. Die zentrale Integrationsaufgabe wird durch *Unifikation* ausgeführt, indem bestimmte feature structures als attribute *constraints* (Einschränkungen, Bedingungen) darstellen, welche verschiedene ihrer Attributwerte untereinander in Bezug setzten. Bei der Unifikation werden nun alle möglichen typgleichen feature structures eingesetzt und überprüft ob die constraints erfüllt sind. Die constraints und die Festlegung der feature structure Typen sind also im Prinzip die Integrationsregeln. Dieser Formalismus erlaubt eine gemeinsame Repräsentation von Integrationseinheiten und -regeln und setzt insbesondere durch mögliche Ersetzungstypen die Bedeutung der multimodalen Perzepte zueinander in Bezug.

Prinzipiell versuchen alle diese Ansätze, sprach-gestische Ausdrücke in Ermangelung einer eindeutigen multimodalen Grammatik mit Hilfe einer — wenn auch nicht immer expliziten — Regelmodellierung aufzulösen. Die Regeln beziehen sich häufig auf temporale, teilweise auch auf semantische (typbezogene) oder seltener auf kontextabhängige Zusammenhänge und Zustände. Mit den Regeln wird versucht, *Struktur* in multimodalen Äußerungen zu erkennen. Eine feste Grammatik dafür vorzugeben wird a) durch die nicht vorhandene Sequenzialität von Sprache und Gestik und b) durch die nicht definitiv fest zu bestimmende zeitliche Synchronität und damit letztendlich durch die Variabilität sprach-gestischer Äußerungen erschwert. Es ist weiterhin anzumerken, dass die hier betrachteten Formalismen und Modellierungsmethoden von *abgeschlossenen* Einheiten sowohl auf sprachlicher als auch auf gestischer Seite ausgehen. In keinem der untersuchten Ansätze wurde die Möglichkeit einer sich *noch in der Artikulation* befindenden Geste in Betracht gezogen.

5.3.3 ATN-Modellierung

Der im Folgenden geschilderte Ansatz basiert auf einer gemeinsamen syntaktischen und semantischen Analyse zeitgestempelter Perzepte sowohl des Sprach- als auch der Gestenerkenner unter Berücksichtigung des Äußerungskontextes. Er stellt insbesondere a) eine Möglichkeit zur Auswertung von noch in der Bewegung befindlicher Gesten (mimetisch/kinemimisch) für die

Umsetzung einer kontinuierlichen Interaktion zur Verfügung und b) ermöglicht die Auswertung rein sprachlicher, sprachlich-gestischer sowie rein gestischer Äußerungen. Um die verschiedenen multimodalen Zusammenhänge (temporal, modal und kontextabhängig) zu modellieren, wird ein **Augmented Transition Network (ATN)** — eine flexible Art eines Übergangsnetzwerks — in einer *erweiterten* Form eingesetzt. Dieses operiert auf einer Menge von Zuständen (states) und Zustandsübergangsfunktionen an den die states verbindenden Kanten (arcs). Ausgehend von einem Startzustand verzweigen unterschiedliche Bedingungen und/oder Heuristiken über Zwischenzustände zu einem Endzustand. Die aktuelle Position, also der eingenommene Zustand repräsentiert den Äußerungskontext, in welchem die Bedingungen überprüft werden. Diese Bedingungen stellen im Fall der beschriebene multimodalen Integration Wortklassifikationsfunktionen, zeitbasierte Zugriffsfunktionen auf die Gestenattributsequenzen oder anwendungsbezogene Statustests dar. Zusätzlich zum beschriebenen Zustandswechsel können im Zuge dieser Wechsel Funktionen ausgelöst werden, um z.Bsp. die Äußerungen zu analysieren und auszuwerten und/oder den Anwendungszustand zu verändern, etwa um eine Interaktion zu triggern oder zu beenden.

Die einzelnen Worte werden von einem kontinuierlichen Spracherkenner mit Start- und (implizit mit) Endzeitpunkten ihrer Äußerung versehen und in eine Queue gestellt, vom Aufbau ähnlich den Attributsequenzen. Die benötigte Information über die gestische Artikulation wird über Zugriffsfunktionen auf die entsprechenden Attributsequenzen ermöglicht. Es handelt sich um ein **erweitertes ATN**, da die einzelnen states um je einen Zeitstempel *reach* ergänzt werden. Dabei wird reach für einen state S bei Vorhandensein eines lexikalischen constraint auf dessen Anfangszeitpunkt gesetzt, wenn dieser zu S geführt hat. Sei also lexikal.start ein entsprechender atomarer Konstituent, dann gilt S.reach = lexikal.start[2]. Führt ein positiver Attributtest in Abwesenheit eines lexikalischen Tests zum Wechsel, so wird der späteste der getesteten Zeitpunkte aus der entsprechenden Attributsequenz verwendet.

[2] Idealerweise sollte hier S.reach auf den Endzeitpunkt der dem aktuellen lexikalischen Konstituenten vorangegangenen Konstituenten gesetzt werden.

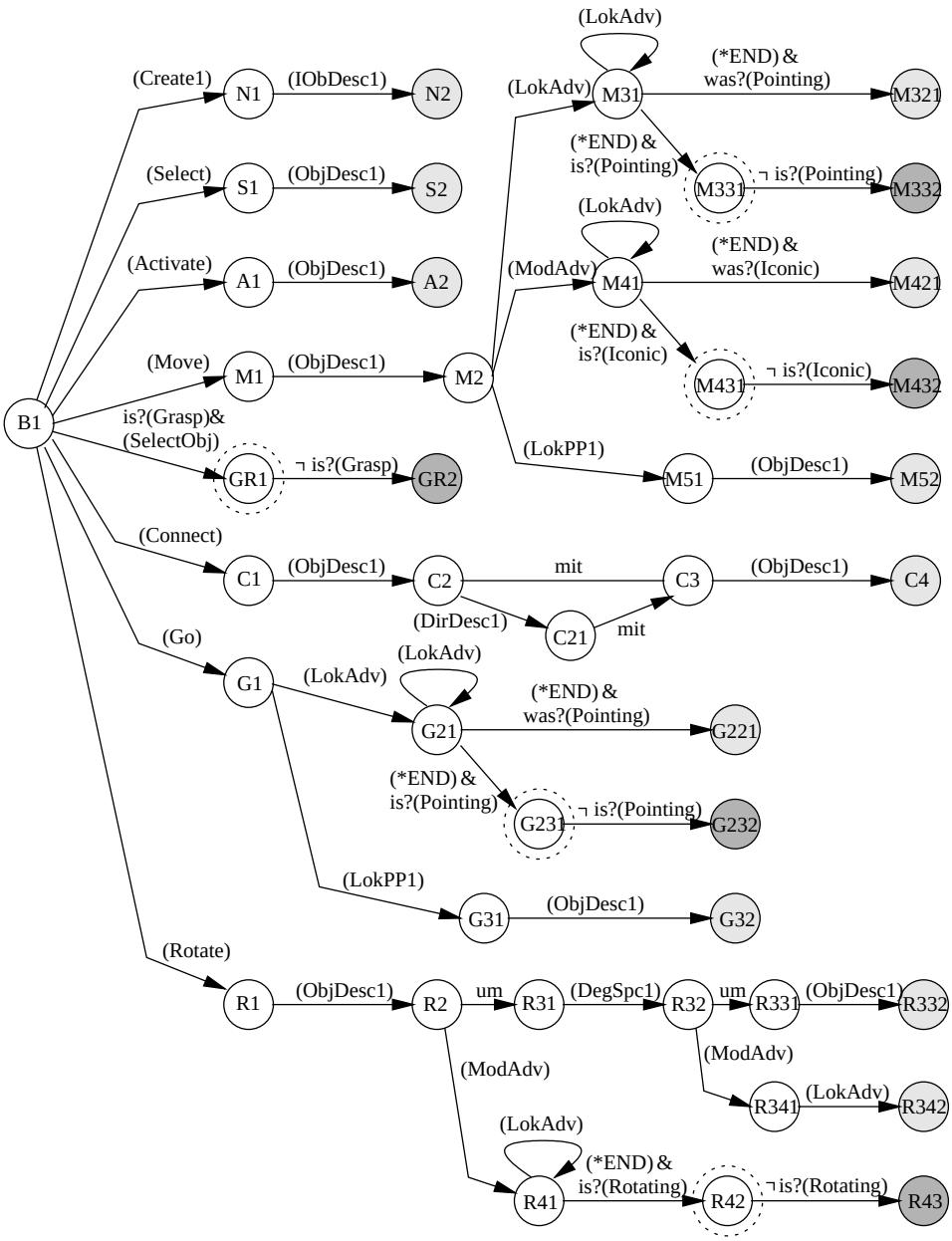

Abbildung 5.1: Augmented Transition Network (ATN) zur integrierten konzeptionellen Darstellung und zur funktionalen Beschreibung des Parsingvorgangs von sprach-gestischen Äußerungen. Zur näheren Erläuterung der Notation und Funktionsweise s. Text sowie Definition 13 auf S. 150

Abbildung 5.1 (S. 146) zeigt ein ATN für die Verarbeitung sprach-gestischer Eingaben. Durch dieses ATN werden die meisten nach Abschnitt 2.3.2 benötigten Interaktionen mit Hilfe einer multimodalen Schnittstelle modelliert. In der Abbildung wurde unter Berücksichtigung der Lesbarkeit weitgehend auf die Hinzufügung der mit einem erfolgreichen state-Übergang verbundenen Analysefunktionen(en) verzichtet. Allein die im Folgenden besprochenen lexikalischen Testfunktionen und die Attributsequenz-Zugriffsfunktionen wurden ergänzt.

In den ATN-Abbildungen wird folgende Notation verwendet: Direkte lexikalische Einträge stehen in Reinform. Vergleiche dazu die Übergänge C2→C3 oder R2→R31 in Abbildung 5.1. Einfach geklammerte Ausdrücke repräsentieren ausgehend vom aktuellen state einen constraint, welcher den durch die Kante dargestellten Übergang ermöglicht. Ausgehend von B1 setzen z.Bsp. unterschiedliche Ereignisse die bereits erläuterte Subkategorisierung um. (Create1) an der Kante B1→N1 bedeutet die Äußerung von Objekterzeugungswünschen, etwa die Worte *"Gib mir, Erzeuge"* etc.B1→GR1, (Grasp) & (SelectObj), prüft dagegen auf das Vorhandensein eines Zugreifens und eines bereits selektierten Objekts respektive, eine rein gestische Aktionsauslösung also. Beginnt ein Ausdruck mit einem *, wie in (*END), so ist damit ein Vorgriff — ein look-ahead — bezeichnet. Im gegebenen Beispiel wird geprüft, ob als nächste Information im Verbalkanal ein *und* oder ein (EOS)[3] folgt. Für viele verbale Konstituenten wird eine weitgehend eindeutige Kurznotation in Anlehnung an die grammatikalische Bezeichnung verwendet. So bedeutet (LokAdv) z.Bsp. einen Test, ob ein Lokaladverb geäußert wurde usw.Geklammerte Ausdrücke, welche auf einen nummerischen Wert enden (wie im Beispiel (Create1)), repräsentieren eine Verzweigung in einen Subgraphen. Abbildung 5.2 veranschaulicht dieses an der Spezifikation des Top-Level-ATN's für die Interaktionsauswertung.

[3] End Of Sentence (Satzende). Dieses wird von der Spracherkennung geliefert (und wird im konkreten System über einen Abfall der Sprachenergie sowie eine explizite Modellierung der Sprachpause erkannt).

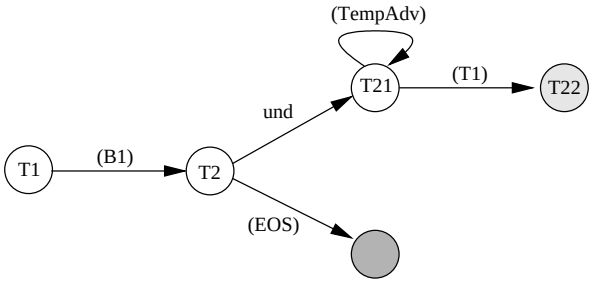

Abbildung 5.2: Das Top-Level-ATN für die Interaktionsauswertung. Die rekursive Gestaltung über die (T1)-Subgraphtraversierung erlaubt die Auswertung zusammengesetzter Interaktionen.

Weiß dargestellte Knoten sind einfache Anfangs- oder Zwischenstates. Die hellgrauen Knoten entsprechen den Vezweigungsknoten in das aufrufende oder aufzurufende Teil-ATN. Ein dunkler Knoten entspricht dem Interaktionsende. Die weißen Knoten mit Punktwolke repräsentieren die Knoten, von denen aus ein Wechsel in den kontinuierlichen Modus der Interaktion stattfindet. In diesem Modus wird die gewünschte Manipulation rein mimetisch/kinemimisch vorangetrieben. Das Ende der kontinuierlichen Interaktion wird allein durch das Beenden der entsprechenden Geste ausgelöst, welche die Interaktion im Vorfeld eingeleitet hat.

Die Integration von komplementären sprach-gestischen Äußerungen (s.a. Abschnitt 2.2.3) erfolgt an Übergangskanten mit konjungierten constraints sowohl auf lexikalischer als auch auf gestischer Basis, z.Bsp. an der Kante M31→M321. Ein erfolgreicher Übergang an solchen Kanten kann nur durch die Erfüllung beider Bedingungstypen geschehen. Die Festlegung der Interaktion wird in Abbildung 5.1 bis auf eine Ausnahme nicht durch unimodales gestisches Artikulieren ausgelöst, sondern durch eine sprachliche Äußerung initiiert. Die Ausnahme stellt B1→GR1 dar, um so die Möglichkeit der Einbindung von Informationen über den aktuellen Anwendungskontext zu demonstrieren: Wurde ein Objekt in einer vorhergehenden Interaktion ausgewählt (selektiert), so reicht nun ein einfaches Zugreifen, um eine drag-and-drop Interaktion anzustoßen.

Zu einer näheren Erläuterung der Funktionsweise wird in Abbildung 5.3 der Bearbeitungsalgorithmus für die Traversierung des erweiterten ATN angegeben. Dieser unterscheidet sich von dem eines rein unimodalen ATNs[4] durch **a)** das Betrachten von parallelen Eingaben und **b)** eine kontinuierliche

[4]Der ATN-Formalismus war ursprünglich für die Bearbeitung sequenzieller, rein sprachlicher Eingaben eingesetzt worden.

Bearbeitung in festen Rasten gemäß einer externen Synchronisationsquelle — gegeben durch die framesynchrone Abarbeitung innerhalb der Applicationstage.

```
set s := startzustand;      ;;Initialisiere state auf Startzustand
for every newframe do       ;;Framesynchrone Ausführung
    lexqueue.update();      ;;Neue lexikalische Einträge in die Queue
    actuators.update();     ;;Aktuatoren (und damit Detektoren) updaten
    set EOP := false;       ;;EndOfParse initialisieren
    do until EOP            ;;Wiederhole solange es etwas zu parsen gibt
        set EOP := true;
        set newstate := false;    ;;Flag für vorzeitigen Abbruch
        ;;Zuerst alle constraints mit lexikalischem Eintrag testen
        for every arc ∈ s.arcs ∧ haslex(arc.constraints) do
            if arc.constraint do            ;;constraints erfüllt?
                set s := arc.tostate;       ;;Traversiere den arc, advance state
                set arc.activate();         ;;Aktiviere Auswertefunktionen
                set newstate = true;
                set EOP = false;
                exit for                    ;;Vorzeitiger Abbruch
            done
        done
        if newstate = false do              ;;Ist vorzeitiger Abbruch?
            ;;Jetzt alle übrigen constraints testen
            for every arc ∈ s.arcs ∧ ¬haslex(arc.constraints) do
                if arc.constraints do
                    set s := arc.tostate;
                    set s.activate();
                    exit for                ;;Vorzeitiger Abbruch
                done
            done
        done
    done                    ;;Ende von do until EOP
done                        ;;Ende der Bearbeitung für diesen Frame
```

Abbildung 5.3: ATN Traversierungsalgorithmus für die Synchronisation des erweiterten ATN mit einer externen Quelle. Im Falle eines VR-Einsatzes entspricht dieses der Applicationstage innerhalb der Render-Loop. Erklärende Kommentare sind eingefügt.

Eine nicht als look-ahead gekennzeichnete lexikalische Abfrage schiebt dabei die lexikalische Queue einen Eintrag weiter. Die Abfragefunktionen is?(<Geste>) und was?(<Geste>) (s. Abbildung 5.1 S. 146) der Attributsequenzen definieren sich gemäß der reach-Zeitpunkte der Zustände zwischen denen sie eine Verbindung herstellen, sowie gemäß der aktuellen Framezeit t_0^e und den damit begründeten Auswertezeitpunkten $\sigma(t_0^e)$ der Aktuatoren und damit der Attributsequenzen.

Definition 13 (Sequenz-Abfragefunktionen)

Sei

- S1 und S2 zwei aufeinander folgende Zustände des erweiterten ATN
- $T_{eval}(t) = \{t_k^e < t_{k+1}^e < \ldots < t_0^e \leq t, t_i^e \in \mathbb{N}_0, k \in \mathbb{Z}_0^-\}$ eine zeitabhängig wachsende endliche Menge von diskreten, disjunkten und geordneten Traversierungszeitpunkten im Ablauf der Render-Loop

So gilt:

Existiert ein weiterer lexikalischer constraint zwischen S1 und S2, dann sei

is?(<Geste>) := HOLDS?(<Geste>,[S2.reach,t_0]),
mit $t_0 \in \sigma(t_0^e)$.

Existiert kein weiterer lexikalischer constraint zwischen S1 und S2, dann sei

is?(<Geste>) := HOLDS?(<Geste>,[t_x,t_0]),
mit $t_x, t_0 \in \sigma(t_0^e) \wedge t_x = min(\sigma(t_{-1}^e) \cap \sigma(t_0^e))$.

Existiert ein weiterer lexikalischer constraint zwischen S1 und S2, dann sei

was?(<Geste>) := $\exists t_x, t_0 \in \sigma(t_0^e)$:
$t_x < t_0 \wedge HOLDS?(< Geste >, [S1.reach, t_x])$
$\wedge \neg HOLDS?(< Geste >, [S1.reach, t_0])$

Mit diesen Defintionen lassen sich nun ebenfalls die constraints, welche auf Attributsequenzen basieren, auswerten. Auf die nähere Angabe der Abfragen des Anwendungskontexts (wie (SelectObj)) wird hier nicht näher eingegangen. Diese stellen quasi statische Abfragen dar, welche die Abfolge in der

Anwendungslogik repräsentieren. In den folgenden Abschnitten wird auf die Auswertung des bisher noch nicht weiter erläuterten (ObjDesc1) Teil-ATN eingegangen, um an der Auswertung der Referenzsemantik die grundsätzlich zu berücksichtigenden Kriterien einer multimodalen Interpretation zu verdeutlichen.

5.4 Äußerungsinterpretation und -umsetzung

Die vorangegangenen Abschnitte haben sich vor allem mit der Frage beschäftigt, wie die Zusammenhänge in der Äußerungsstruktur sprachlicher und gestischer Perzepte definiert sind und welche Form der Repräsentation dieser Zusammenhänge sinnvoll ist. Dabei wurde das Konzept des erweiterten ATN vorgestellt und wesentliche Teilbereiche (s. die Abbildungen 5.1 und 5.2) für den Einsatz in einer Interfacekomponente für multimodale Interaktionen ausgearbeitet. In diesem Zusammenhang stellt die Einbindung auswertungsbezogener interpretativer Funktionen bei einer Traversierung innerhalb eines ATN die Möglichkeit zur Verfügung, die Äußerungen im Verlauf zu analysieren und ggfs. eine Anwendungsreaktion auszulösen. In diesem Abschnitt wird nun ein Verfahren ausgearbeitet, um die für eine Interaktionsangabe unbedingt notwendigen Objekte zu identifizieren — eine multimodale Referenzanalyse.

5.4.1 Ein Ansatz zur multimodalen Referenzanalyse

Um eine integrierte Interpretation gestischer und sprachlicher Deixis — der Information über kommunizierte Referenzen — zur Verfügung zu stellen, wird an dieser Stelle ein auf schrittweisen Mengenoperationen basierendes Konzept entwickelt. Dieses beruht auf sprachlicher Seite auf der Analyse der DNPs. Auf gestischer Seite berücksichtigt es die Betrachtung der Kopf- und Handrichtungen zum Zeitpunkt der Äußerung der einzelnen DNP-Konstituenten sowie die Untersuchung des Zeigeverhaltens während der Äußerung. Abbildung 5.4 illustriert ein entsprechendes erweitertes ATN für das Parsen der DNPs und gibt die entsprechenden Traversierungsstates an.

Dieses Vorgehen ist geleitet durch die Annahme, dass die Ermittlung der entsprechenden Referenten bei einem Rezipienten als eine Art Suchprozess geschieht, bei welchem die Menge der in Frage kommenden Objekte durch jede neu hinzukommende Information weiter eingeschränkt wird. Dabei werden sowohl sprachliche als auch gestische Informationen in einem einheitlichen Prozess einbezogen. Die gestische Deixis definiert durch ihre eindeutige Richtung *Halbräume* (s.n. Abschnitt) im umgebenden Raum. In

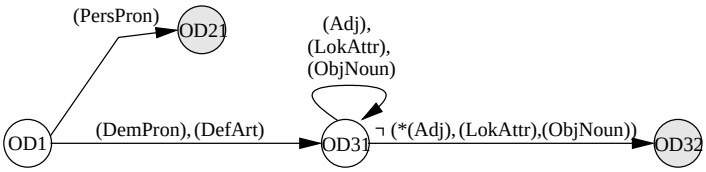

Abbildung 5.4: Teil-ATN für die Auswertung von Objektbeschreibungen in Form von DNPs oder Personalpronomen. Das Kürzel OD steht für die Bezeichnung ObjDesc in Abbildung 5.1 auf S. 146.

diesem liegen die Objekte, welche im Kontext der DNP in Frage kommen[5]. Als Annahme über direkt vermittelte Referenzen steht hier die Voraussetzung, dass das entsprechende Objekt im Laufe der Äußerung mindestens im Betrachtungsraum des Interagierenden lag. Über den Zeitraum einer DNP-Äußerung werden also die Objekte in der Blickrichtung als Grundlage für die weitere Referenzermittlung benötigt. Jede weitere verbale aber auch gestische Äußerung schränkt diese Objekmenge durch eine Schnittmengenbildung weiter ein. Als Beispiel soll folgende multimodale Teiläußerung dienen: " ... [Zeigegeste]*dieses gelbe Vierkantrohr* ... ". Die entsprechende Traversierungsabfolge des in Abbildung 5.4 angegebenen ATN ist dann wie folgt: OD1 $\rightarrow$ OD31 $\rightarrow$ OD31 $\rightarrow$ OD32. Für die gelingende Angabe des Interpretationsalgorithmus müssen zuvor folgende Vereinbarungen getroffen werden, es sei:

- $\mathcal{B}(i)$ die Menge aller im Intervall i angeschauter Objekte.
- $\mathcal{Z}(i)$ eine absteigend sortierte Menge von Objekten. Deren Sortierung ist ein Ausdruck für die Güte mit der bestimmt werden kann, ob auf das Objekt im Intervall i gezeigt wurde.
- $\mathcal{O} \circ \mathcal{A}$ eine Verknüpfung zur Bestimmung der Menge von Objekten aus $\mathcal{O}$, für die die Attribute aus $\mathcal{A}$ gültig sind.
- $ISATTRIB?(< state >)$ eine Testfunktion: Bezeichnet der lexikalische Konstituent, welcher zum state geführt hat, ein Attribut?
- $HASATTRIB?(< object >, < state >)$ ein Test auf die Gültigkeit des zum state führenden Attributs für ein Objekt.

[5]An dieser Stelle werden anaphorische Konstrukte nicht berücksichtigt. Dieses muss ggfs. auf einer weiteren Analyseebene geschehen.

Bezeichne $\mathcal{A}_{DNP}$ die Menge aller im DNP-Äußerungsverlauf vorkommender Attribute, so lässt sich eine Menge $\mathcal{R}$ möglicher Referenten bestimmen durch:

$$\mathcal{R} = (\mathcal{B}(i) \circ \mathcal{A}_{DNP}) \cup \mathcal{Z}(i) \tag{5.1}$$

```
begin proc OD1.onEnter()                          ;;Für jeden arc → OD1
    set states := ∅;                              ;;State-Speicher initialisieren
    set states := states + OD1;                   ;;Den aktuellen state merken
    for every arctravers do                       ;;Für jede Traversierung im DNP
        set states := states + newstate;          ;;Den neuen state merken
        if newstate = OD31 then do                ;;Ist es letzter DNP-state?
            ;;Das Auswerteintervall des DNP bestimmen
            set i := [OD1.reach − offset, OD32.reach];
            set objects := B(i);
                for every state ∈ states do       ;;gespeicherte Attributstates testen
                    if state = OD31 & ISATTRIB?(state) then do
                        ;;Die erste Mengenschnittbildung umsetzen
                        for every obj ∈ objects do
                            if ¬HASATTRIB?(obj, state) then do
                                set objects := objects − obj;
                            done
                        done
                        ;;Ergebnis der parallelen Gestenauswertung anfordern
                        set pobjects := Z(i);
                        ;;Mit dem Ergebnis der Gestenauswertung verbinden
                        if pobjects = ∅ then do
                                return(objects);
                        done
                        ;;Die zweite Mengenschnittbildung umsetzen
                        for every obj ∈ pobjects do
                            if ¬(obj ∈ objects) then do
                                set pobjects = pobjects − obj;
                            done
                        done
                            return(pobjects);
                    else if state = ... ; then do
                    ⋮
    ⋮
end proc
```

Abbildung 5.5: Der Auswertealgorithmus der Referenzanalyse. Die über dem Äußerungsintervall betrachteten Objekte werden zuerst gegen geäußerte Attribute getestet und daraufhin mit einer Menge von gestisch bezeigten Objekten geschnitten.

Der in Abbildung 5.5 wiedergegebene Algorithmus ist eine direkte Umsetzung der in Gleichung (5.1) angegebenen Heuristik zur Referentenbestimmung mittels Mengenoperationen. Der Algorithmus geht folgendermaßen vor: Im Verlauf des DNP-Parsing erfolgt für eine abschließende Auswertung eine Zwischenspeicherung der einzelnen besuchten states. So kann zuerst das gesamte Äußerungsintervall i vom Erreichen des OD1-state bis zum Erreichen des OD32-state bestimmt sowie daraufhin die Menge $\mathcal{B}(i)$ aller im Blick gelegener Objekte für die Auswertung zugrunde gelegt werden. Der angegebene Offsetwert (s. OD1.reach-offset) ermöglicht als Parametrisierung eine Ausdehnung des betrachteten Intervalls. Bei Erreichen des letzten DNP-states werden nun die Eigenschaften für jedes Objekt dieser Menge mit den im Verlauf geäußerten Attributbeschreibungen verglichen. Stimmen die Attribute nicht überein, so wird das entsprechende Objekt aus der Menge entfernt.

In einem zweiten Schritt wird überprüft, ob während des Intervalls i auf Objekte gezeigt wurde. Wenn dieses zutrifft, dann liefert die Gestenauswertefunktion $\mathcal{Z}(i)$ eine sortierte Liste von Objekten zurück. Jetzt gilt es noch, die Schnittmenge zwischen der *Gestenliste* und der Objektmenge unter Erhalt der Sortierung zu gewährleisten. Die resultierende Liste gibt in absteigender Reihenfolge mögliche Interpretationen der multimodalen Gesamtäußerung und damit mögliche Referenten zurück. Die ATN-Modellierung ermöglicht die einfache Umsetzung des gegebenen Algorithmus durch die Anreicherung der benötigten Funktionen. Arc-spezifische Funktionen werden direkt am entsprechenden arc notiert, state-spezifische Funktionen werden an jedem zum state führenden arc notiert. Gemäß Abbildung 5.1 erfolgt bei einer Traversierung die entsprechende Funktionsausführung (s. Funktion $arc.activate()$ in Abbildung 5.3 auf S. 149).

5.4.2 Auswertung gestischer Referenzinformation

Die bisherigen Ausführungen zum Auswertevorgang deiktischer Äußerungen haben die gestische Seite der Interpretation durch die Angabe der "High-level"-Funktionen $\mathcal{B}(i)$ und $\mathcal{Z}(i)$ eingebunden. Offensichtlich tritt hier im Zusammenhang eines VR-Einsatzes eine Schwierigkeit zutage: Die Aufgabe der gestischen Auswertung in Kombination mit dem sprachlichen Gebrauch definiter Nominalphrasen besteht in der Bestimmung von angeschauten und evtl. durch Zeigegesten bestimmten möglichen Referenten. Die eigentliche Interpretation findet aber erst *nach* erfolgter Äußerung aller Konstituenten mit deiktischem Bezug statt, welches ohne weiteres bis zu zwei Sekunden oder länger dauern kann. In dieser Zeit ist es einem in einer immersiven

Anwendung (durch den Einsatz von Head-Tracking und Stereoskopieverfahren) operierenden Anwender ohne weiteres möglich, unterschiedliche Blickperspektiven auf eine Szene einzunehmen — genau wie es während einer Kommunikation in einem realen Szenario vorkommen kann. In den eingesetzten VR-Systemmodellierungen stehen nur Informationen für den *aktuellen* Renderschritt oder Frame zur Verfügung. Durch die Kopplung mit der Render-Loop werden zu diesem Zeitpunkt ebenfalls die möglichen neuen states des ATN traversiert. Die damit verbundenen Auswertungen benötigen aber Informationen aus dem gesamten *zurückliegenden* Auswerteintervall. Wie kann in diesem Fall die *indexikalische Semantik* der einzelnen Äußerungen garantiert werden?

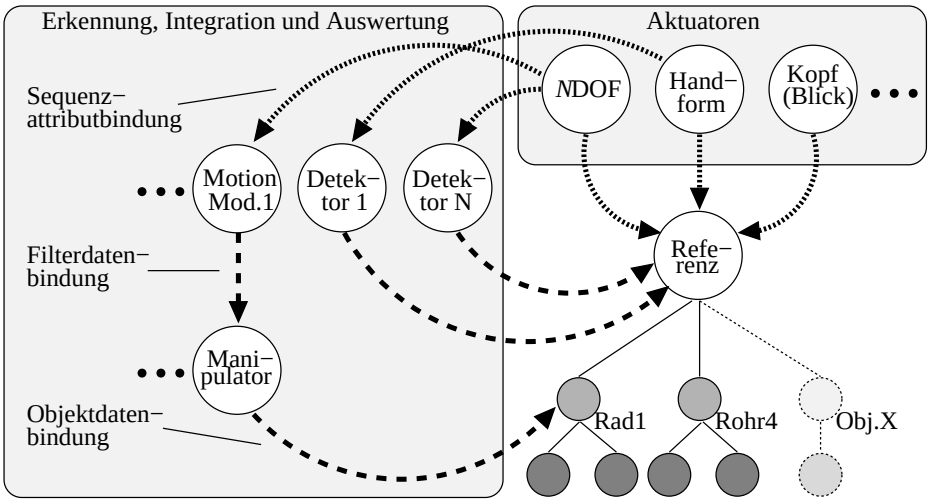

Abbildung 5.6: Einbettung der verschiedenen Komponenten zur sprachgestischen Auswertung in einer VR-Anwendung: Attributsequenzen dienen dem Informationsaustausch zwischen Aktuatoren, Detektoren, Motion-Modifikatoren und Referenzknoten. Simulationszeit-adäquate Teilanweisungen werden über die Motion-Modifikatoren und Manipulatoren auf entsprechende Szenengraphknoten angewendet.

Der hier vorgeschlagene Weg setzt zur Lösung dieser Aufgabe eine kontinuierliche und mit der wechselnden Perspektive schritthaltende Auswertung der über den Kopf und die Hände zu ermittelnden indexikalischen Informationen ein. Für einen möglichst allgemeinen VR-Einsatz wird dafür eine weitere spezielle Knotenklasse in den AS-Graphen eingeführt, der *Referenzknoten*. Dieser bedient sich zum einen der Daten über die Benutzerextremitäten und die ausgeführte deiktische Gestik, welche abstrakt mittels der

Aktuatoren und Detektoren über Sequenzattribute im VR-Kontext etabliert werden (s. Abbildung 5.6), zum anderen hat dieser Knoten Wissen über die in der virtuellen Szene befindlichen Objekte und deren Attribute. Die für diese Aufgabe in Frage kommenden Aktuatortypen wurden bereits in den Abschnitten 4.2.4 und 4.2.4 konzipiert und ausgearbeitet.

Nun wird zu jedem Renderschritt eine Abstandsberechnung der Szenenobjekte bezüglich der durch die Aktuatoren verankerten Referenzstrahlen durchgeführt. Durch die Einführung eines *parametrisierten* Grenzwinkels (α und β) wird weiterhin eine Möglichkeit geschaffen, die Form der gebildeten Halbräume im Verlauf zu beeinflussen. Die Teilräume können z.Bsp. durch Informationen über die Gestenexpression — etwa durch Wissen über die Gestenklimax u.Ä. — einen kegel- oder konusförmigen Bereich für die Aktuatoren weiter *fokussieren*. So wird die Menge der in Frage kommenden Objekte verringert und gleichzeitig eine Möglichkeit geschaffen, die Güte der Auswahl zu verbessern. Dieses Konzept realisiert ein parametrisierbares *Culling* — eine Auswahl, welche adaptiv außerhalb des Cones liegende Objekte ignoriert. Innerhalb liegende werden gemäß ihres Strahlabstandes (hier dem Verhältnis von d_m zu d_r und dem Abstand zum Strahlursprung) sortiert.

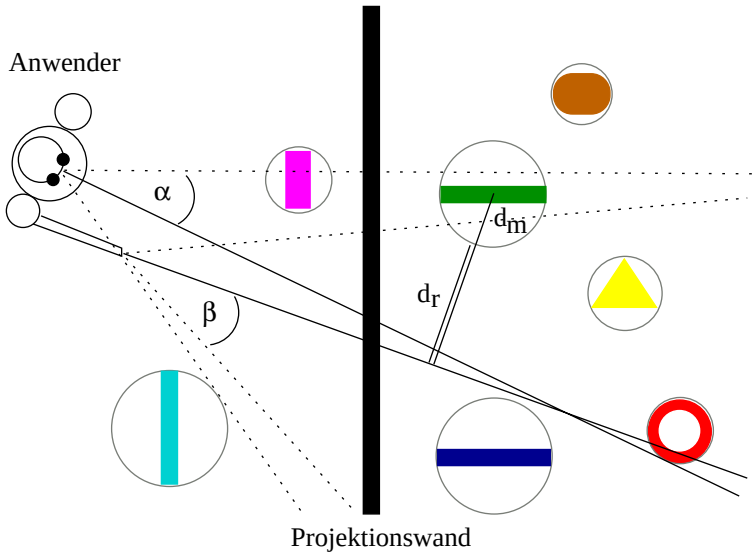

Abbildung 5.7: Fokus zweier primärer Referenzstrahlen und exemplarische Abstandsermittlung in einer virtuellen Szene.

In Abbildung 5.7 wird ein schematischer Ausschnitt während der Interaktion in der Aufsicht skizziert. Man erkennt dort zwei aktive Segmentierungen

des virtuellen Raums gegeben durch die Blickrichtung und die Richtung des rechten Arms eines Benutzers. In der Abbildung finden bereits *degenerierte* Halbräume Verwendung, um die Anzahl der in Frage kommenden Objekte weiter einzuschränken. Zwei verschiedene Abstandsmaße der Objekte zu den Primärstrahlen ermöglichen eine genauere Sortierung dieser Objekte gemäß des aktuellen Abstands. In der Abbildung wird der Abstand zum Objektmittelpunkt sowie der Abstand zu der ein Objekt umgebenden Sphäre skizziert. Die gepunkteten Linien veranschaulichen den Fokusbereich von zwei Referenzstrahlen. d_m und d_r notieren exemplarische Abstandsermittlungen zwischen dem durch den NDOF-Bewegungsaktuator der rechten Hand gelieferten Primärstrahl zum Mittelpunkt und zum sphärischen Umfang eines Objekts respektive.

Deiktisches Mapping mit Hilfe von Spacemaps

Die gewonnen Informationen über Objektordnungen werden nun für jeden Frame und jeden Referenzstrahl gemäß einer Abstandsfunktion f_{dist} in Listen L' bereitgestellt. Die Listen über einem Zeitintervall i werden zusammengefasst und bilden eine *Spacemap* für das entsprechenden Zeitintervall.

Definition 14 (Spacemap) *Eine Spacemap S repräsentiert eine Menge von diskreten — durch Güteinformationen angereicherten — Objektsortierungen gemäß einer Objektordnungsfunktion bezüglich eines Referenzstrahls für diskrete Zeitpunkte in einem Intervall i.*

$$S(t,i) := \{L(t_a^e), L(t_{a+1}^e), \ldots, L(t_e^e), t_a \leq t_a^e \leq t_e^e \leq t_e \leq t, t_x^e \in T_{eval}(t), \\ i = [t_a, t_e], L(t_x^e) = (< info >, L'(\mathcal{O}(t_x^e), f_{dist}))\} \tag{5.2}$$

Dabei sei

- $T_{eval}(t) = \{t_k^e < t_{k+1}^e < \ldots < t_0^e \leq t, t_i^e \in \mathbb{N}_0, k \in \mathbb{Z}_0^-\}$ eine zeitabhängig wachsenden endliche Menge von diskreten, disjunkten und geordneten Simulationsschritten, den Traversierungszeitpunkten im Ablauf der Render-Loop.
- $\mathcal{O}(t_x^e)$ die Menge der Objekte in der Szene zum Zeitpunkt t_x^e.
- $L'(\mathcal{O}(t_x^e), f_{dist})$ die Menge der gemäß f_{dist} sortierten Objekte zum Zeitpunkt t_x^e.
- $< info >$ eine Informationsstruktur mit zusätzlicher Gütebewertung.

Dieser Abbildungsvorgang von den Objektpositionen im Raum auf eine Ordnung in Bezug auf die ebenfalls im Raum lokalisierten deiktischen Informationen der Aktuatoren wird hier in Anlehnung an das *ikonische Mapping* (s. Abschnitt 2.1) als **deiktisches Mapping** bezeichnet. Die in den Spacemaps verankerte Information ermöglicht nun über den Referenz-Knoten Anfragen über die Ordnung der Objekte gemäß des entsprechenden Referenzstrahls zu einem konkreten Zeitpunkt t_x oder über einem Intervall i. Sei $guete(< info >, t_x^e)$ der Gütewert zum Zeitpunkt t_x^e, so werden $\mathcal{B}(i)$ und $\mathcal{Z}(i)$ auf Basis dieser Spacemaps z.Bsp. definiert als:

$$\mathcal{B}(i) := \bigcup_{t_x^e \in i} L(t_x^e) \tag{5.3}$$

$$\mathcal{Z}(i) := L(t_y^e), \text{mit} \nexists\, t_x^e \in i : guete(< info >, t_y^e) < guete(< info >, t_x^e) \tag{5.4}$$

Die an den Referenzknoten gerichtete Abfragefunktion kann dabei so grundlegend wie die Rückgabe des ersten Objektes, einer vollständigen Liste oder von mehreren Listen sein (z.Bsp. bei mehrfach vorkommenden betrachteten gleichen Gestentypen im Intervall i).

Abbildung 5.8 veranschaulicht die "Wanderung" eines Objekts, dargestellt durch das ausgefüllte Oval, in einer Spacemap während des Vorgangs des Fokussierens dieses Objekts. In 5.9 wird dieser Vorgang für das Aktuatortrio rechte Hand, linke Hand und Blickrichtung verdeutlicht. Die Inhalte der exemplarischen $< info >$ Einträge bedeuten dabei folgendes: FT ist der aktuelle Zeitpunkt des Frames, also FT=t_x^e. Hint ist eine Güteeintrag zur Bewertung einer evtl. während der Listenerstellung geäußerten deiktischen Geste — im dargestellten Beispiel eines Zeigens. Raw dagegen ist der entsprechende zugrundeliegende Wert des Referenzstrahls.

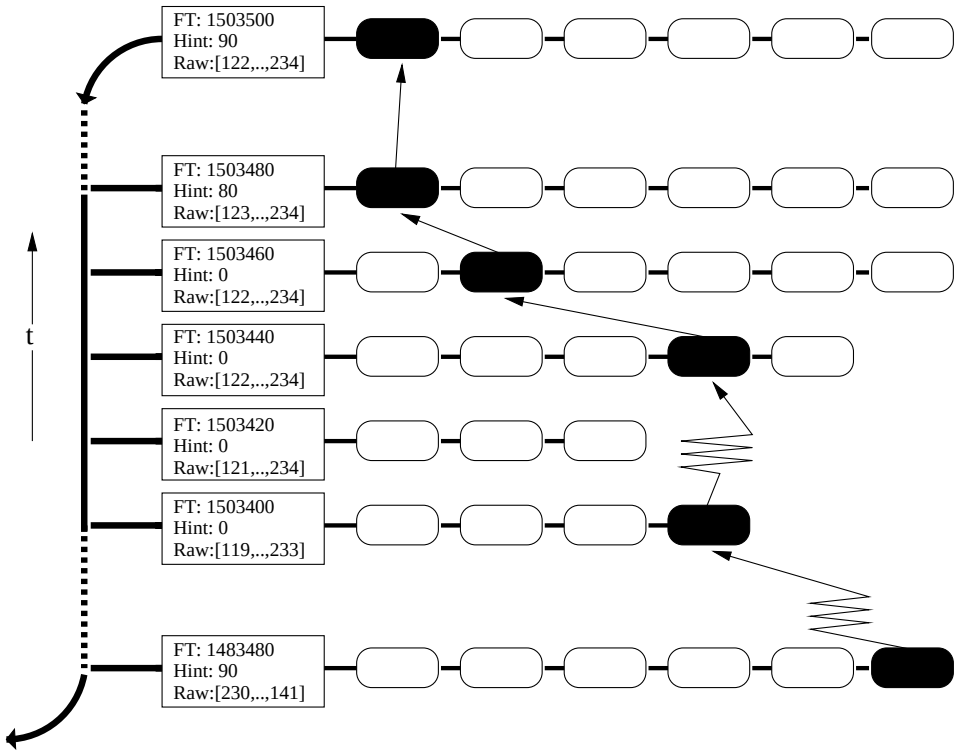

Abbildung 5.8: Spacemap für einen Referenzstrahl

Auf konzeptueller Ebene wird mit der Einführung dieser Referenz-Knotenklasse die Möglichkeit zur rückschauenden Auswertung gestisch-deiktischer Referenzen geschaffen. Durch die Wahl einer kontinuierlichen Indexauswertung für jedes neu berechnete Bild wird eine Unabhängigkeit insbesondere von dynamischen Szenen — etwa mit bewegten Objekten und anderen Simulationsveränderungen — gewährleistet. Eine zeitliche Zuordnung kann in den Spacemaps immer rekonstruiert werden. Über die Art der gewählten Vorsortierung f_{dist} zu jedem Frame wird an dieser Stelle keine entgültige Annahme gemacht. Das testweise ausgearbeitete und implementierte Abstandverfahren unter Berücksichtigung des Mittelpunkt- und Umfangabstands eines Objekts stellt hier einen ersten Vorschlag zur Realisierung und Erläuterung des Verfahrens dar. So bietet die Konzeption die Möglichkeit, auch an dieser Stelle Einfluss auf den Auflösungsvorgang zu nehmen. Experimentelle Ergebnisse aus Aufmerksamkeitsstudien über den Betrachterfokus oder über das zeitliche Verhältnis von verbalen Äußerungen visueller Attribute zu dem Zeitpunkt ihrer Betrachtung, können in der hier erfolgten Konzeption ebenso berücksichtigt und modelliert werden.

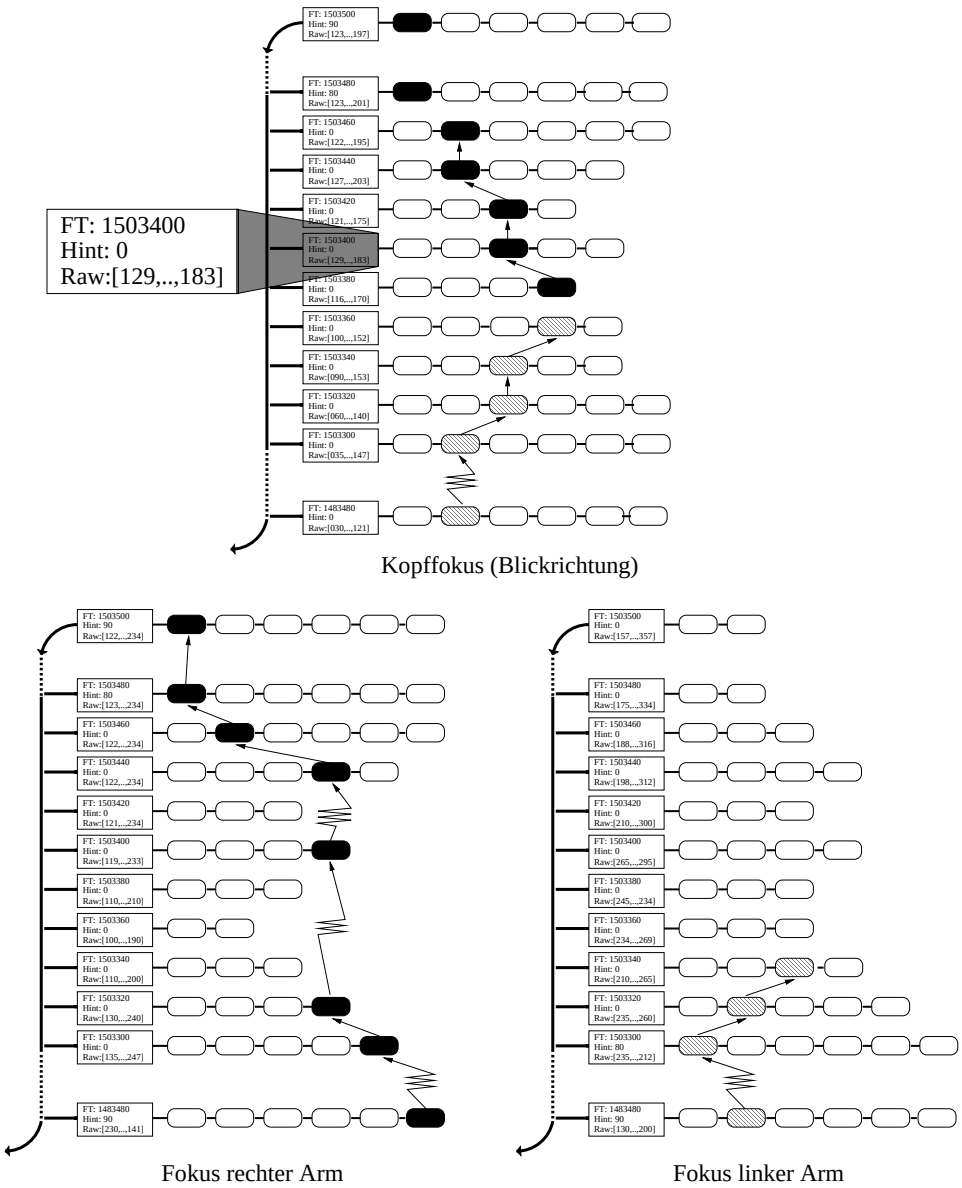

Abbildung 5.9: Spacemaps für die drei primären Referenzstrahlen der rechten Hand, der linken Hand und der Kopf-/Blickrichtung während der Fokussierung eines Objekts.

5.4.3 Manipulationsumsetzung

Für die Umsetzung der gewünschten Interaktionsschritte stehen nun sämtliche benötigten Komponenten und Funktionen zur Verfügung. Über die Subkategorisierung an den vom B1-state fortlaufenden Kanten des in Abbildung 5.1 auf S. 146 angegebenen ATN wird der Typus der Interaktion bestimmt. Das Erreichen der einzelnen n-States führt ggfs. die multimodalen Informationen zusammen. Die in den vorangegangenen Abschnitten ausgearbeiteten Verfahren liefern die für die Interaktion benötigten Objekte. An entsprechenden Übergängen (z.Bsp. R42 oder M331, s. Abbildung 5.1) entscheidet sich die Art der Interaktion, ob sie also in einem diskreten Schritt ausgeführt werden soll oder ob eine mimetisch/kinemimisch vermittelte kontinuierliche Interaktion beginnt. Das Erreichen der End-States gewährleistet im ersten Fall durch den Aufbau des ATN, dass alle notwendigen Informationen jetzt zur Verfügung stehen. Es ist einerseits die Interaktionsart bestimmt worden und andererseits garantiert die erfolgreiche Traversierung bis zu diesem Punkt, dass auch die für die Interaktion benötigten Objekte bekannt sind. Eine Umsetzung in der Virtuellen Szene kann erfolgen. Wird dagegen das Ende der Äußerung durch die Spracherkennung mitgeteilt, aber es liegt keine gültiger Endzustand vor, so misslingt die Interaktion und wird daraufhin verworfen.

Startet dagegen eine kontinuierliche mimetisch/kinemimische Interaktion, dann werden die entsprechenden Bindungen zwischen den jeweiligen interaktionsspezifischen Aktuatoren, Motion-Modifikatoren und Manipulatoren aufgebaut. Die gestische Bewegung erhält nun die Kontrolle über die Manipulation. Die Dauer der Bindungen zwischen den Vermittlerobjekten wird allein durch die Testprädikate des entsprechenden Motion-Modifikators festgelegt. Dieses stellt ein Modellierungskonzept für Interaktionen mit Hilfe mimetischer/kinemimischer Gesten unter Beibehaltung eines direkten Feedbacks in der virtuellen Szene zur Verfügung. Die gleichzeitig mögliche Inspektion des Interaktionsresultates lässt damit eine explorative Vorgehensweise zu. In Abbildung 5.10 wird am Beispiel der Rotationsumsetzung exemplarisch der Informationsfluss zwischen Aktuator, Motion-Modifikator und Manipulator im Gesamtschema noch einmal verdeutlicht.

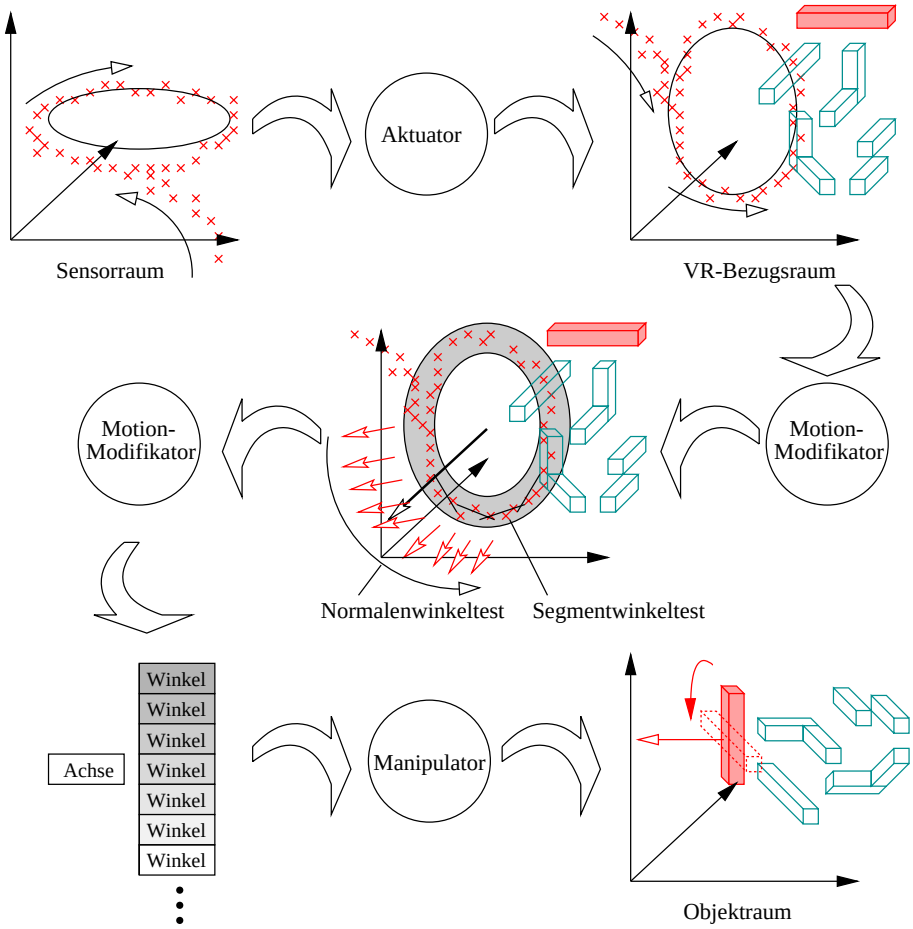

Abbildung 5.10: Beispielhafte Manipulationsvermittlung durch das Zusammenwirken von Aktuator, Motion-Modifikator und Manipulator während einer gestischen Rotationsausführung.

5.5 Zusammenfassung

Der entwickelte ATN-Integrationsformalismus erlaubt es, verschiedene Informationsquellenquellen — hier die Sprach- und Gestenverarbeitung sowie die Zustandsbeschreibung der Anwendung — einheitlich zu repräsentieren. Dabei werden die temporalen und semantischen Zusammenhänge funktional definiert und als Kantenconstraints innerhalb des ATN zur Anwendung gebracht. Entgegen der Event-getriebenen rein seriellen Ausführung des herkömmlichen ATN-Modells, erlauben die entwickelten Erweiterungen eine kontinuierliche Auswertung auch paralleler Eingaben und erfüllen damit die im Vorfeld genannten Voraussetzungen für eine multimodale Integration innerhalb einer VR-Anwendung. Für die hier vorgestellten Interaktionsarten genügt das vorgestellte Verfahren auch in Bezug auf seinen Verwaltungsaufwand. So lassen sich Modifikationen, etwa bei einem Einfügen neuer Constraints oder der Erweiterung des Wortschatzes, problemlos vornehmen.

Die Form der ausgearbeiteten Referenzanalyse betrachtet sowohl die Blick- als auch die Zeigerichtung. Die vorgestellte Version basiert auf drei der vorgesehenen zehn Referenzstrahlen. Die eingesetzte Heuristik für die Objektsortierung mit den zugrundeliegenden mathematischen — vor allen Dingen trigonometrischen — Funktionen, stellt einen guten Kompromiss zwischen Akuratheit und Rechen- bzw. Zeitbedarf der Auswertung dar. Die schritthaltende Auswertung und zeitliche Pufferung in Spacemaps ist im Kontext einer immersiven Anwendung als notwendig anzusehen, um die Gültigkeit bei einer zeitversetzten Analyse zu gewährleisten. Dieses Konzept einer engen Kopplung und vorausschauenden Teilverarbeitung lässt sich dabei vielfältig auch auf die Bearbeitung anderer Gestikformen anwenden. Eine definierte Berechnungsaufgabe im Zusammenspiel mit der Anbindung an die virtuelle Szene liefert die Grundlage, um auch hier übertragbare Konzepte und deren Realisation in Form von Szenengraphkomponenten einzuführen.

Kapitel 6

SGIM - eine exemplarische Anwendung in der Virtuellen Konstruktion

Nach der analytischen Beschreibung der Zusammenhänge zwischen Gestik und Sprache und der daraus resultierenden Begriffs- und Methodenbildung sowie der darauf folgenden Ausarbeitung der benötigten Konzepte und Verfahren, liegt der Schwerpunkt in diesem Kapitel auf dem Implementierungsteil. Das Thema ist die Umsetzung der entwickelten Konzepte in einem realen System — SGIM, einer mit einer multimodalen Schnittstelle ausgerüsteten Anwendung zur Virtuellen Konstruktion. Neben der Beschreibung Virtueller Umgebungen und benötigter Interaktionen speziell des CVK, dem gewählten Zielsystem für die Entwicklung einer beispielhaften multimodalen Eingabe, wurden grundlegende Betrachtungen der Eingabemodalitäten Gestik und Sprache angestellt und daran anknüpfend die Anforderungen an ein technisches Interpretations- und Integrationssystem ermittelt. Die darauf aufbauenden Konzeptionen von Kernkomponenten zur Einbettung multimodaler Interaktionen in VR-Systeme werden hier als Grundlage des Designs und der Implementation einer exemplarischen Anwendung verwendet.

6.1 Aufbau und Funktionsbeschreibung

SGIM erlaubt einem Anwender multimodale, sprach-gestische Interaktion mit dem CVK. Grundobjekte zur Konstruktion eines Citymobils — Vierkantrohre verschiedener Länge, Räder und Verkleidungsteile — dienen als Bauelemente im Konstruktionsvorgang. Ausgelegt ist der SGIM-Prototyp

zum Einsatz vor einer Wall, einem immersiven Großdisplay mit einer Fläche von ca. 2x3m, ist aber darauf nicht beschränkt. Einzelne Komponenten, z.Bsp. die Referenzanalyse, wurden bereits zum Test der Zeigegestenpräzision auf die Workbench mit ihrer ca. 1.5x1m großen Projektionsfläche übertragen. Die Szene wird über Shutterbrillen und elektromagnetisches Headtracking stereoskopisch dargestellt und somit der Benutzer immersiv eingebunden. Sensoren auf dem Kopf und den Händen sowie die verwendeten Datenhandschuhe liefern die benötigten Informationen über die Körperkonfiguration. Über ein Funkmikrophon erfolgt eine Anbindung an eine sprecherunabhängige kontinuierliche Spracherkennung. Alle eingehenden Daten werden von einem zentralen Servermodul zeitgestempelt und daraufhin netzwerkweit versendet, um so eine Entkopplung zwischen Rohdatenaquisition und Interpretation und Anwendung zu gewährleisten.

Eine Hauptfunktionalität im Kontext der VK ist die Etablierung von Verbindungen zwischen Objekten. Dazu können verbindbare Objekte ausgewählt und der Verbindungswunsch umgesetzt werden. Ein multimodales Interface lässt den SGIM-Anwender Objekte instanziieren und die daraufhin dargestellten Teile referenzieren, indem er sie sprachlich beschreibt und dazu "natürlich" gestikuliert, in der Szene herumschaut und dabei *frei* nach Ermessen Zeigegesten ausführt. So ausgewählte Objekte können entweder in Form einer diskreten Interaktion direkt mit anderen referenzierten Objekten über eine Verbindungswunschäußerung verbunden werden (*Nimm* [Zeigegeste] *dieses blaue Rohr und setze es* [Zeigegeste]*da dran.*) oder sie können erst selektiert und anschließend mit Hilfe kontinuierlicher Veränderungen sukzessive manipuliert werden. Hier wurde durch eine Erkennung von Zugreifen und Öffnen der Hand sowohl ein direkt vermitteltes distantes Führen (die Objekte müssen nicht im Greifraum des Anwenders liegen, sondern können auch außerhalb der Benutzerreichweite positioniert sein), als auch die mimetische Beschreibung einer gewünschten Lageveränderung durch gestisches Vormachen des entsprechenden Manipulationsverlaufs ermöglicht. Die erst beschriebene Funktion — das sogenannte *dragging* — kann ebenfalls zu einer Verbindung führen, da der Benutzer während der Interaktion durch sprachliche Anweisung in einen Verbindungsmodus treten kann. In einem solchen Fall registriert eine integrierte Kollisionserkennung etwaige Geometrieüberschneidungen, prüft diese Stellen gemäß der Konstruktionswissenbasis auf mögliche kompatible Ports und etabliert daraufhin ggfs. eine entsprechende Verbindung. Mit diesen Interaktionmöglichkeiten können so sukzessive komplexe Modelle mit Hilfe der multimodalen Eingabe realisiert werden. Die dabei eingesetzten Verfahren bedienen sich der hier erarbeiteten Konzepte, welches im Folgenden am Detail der Implementierung herausgestellt wird.

Neben diesen rein für den Konstruktionsprozess benötigten Funktionen wurden ebenfalls exemplarisch einfache Möglichkeiten der Aktionsauslösung und der Navigation geschaffen. Die Konstruktionsszene ist in das Virtuelle Labor eingebettet. In dieser Umgebung kann sich ein Benutzer zu verschiedenen Orten hinbewegen, indem er z.Bsp. *"... geh zu Spot X."* äußert oder Türen u.Ä.aktiviert. Daneben wird über die Spracheingabe die Sensorik kallibriert, um so die durch den Kabelzug an der Sensorik auftretenden Abweichungen im Verlauf einer längeren Interaktionssitzung auszugleichen.

6.2 Wahl des Modellierungswerkzeugs

VR-Anwendungen basieren wie viele computergraphische Systeme auf der internen Modellierung der virtuellen Welt mit Hilfe von Szenengraphstrukturen (s. Abschnitt 2.3). Darüber hinaus wird bei der Entwicklung allgemeiner VR-Autorensystemen mehr und mehr die portable Modellierung (von Teilen) des Programmablaufs angestrebt. Die Berücksichtigung beider Faktoren war ein wichtiges Kriterium für die Ausarbeitung der zuvor entwickelten Konzepte. Zum einen werden Instanzen der Aktuatoren, Motion-Modifikatoren, Manipulatoren, Detektoren und Referenzknoten entweder direkt in den Szenengraphen eingebettet, da sie relative Transformationsinformationen benötigen, oder sie sind allein über ihre Ablaufsteuerung in die Render-Loop eingebunden, wie es für den Traversierungsalgorithmus des erweiterten ATN notwendig ist. Zum anderen bestehen zwischen den einzelnen Komponenten Datenverbindungen, welche nicht allein mit den Methoden innerhalb der für die Computergraphik und VR typischen Baum-Traversierung abgebildet werden können, da diese Verbindungen orthogonal zu der Szenengraphstruktur verlaufen.

Angestrebte Erweiterungen von existierenden Knotenklassen lassen sich in fast allen existierenden Autorensystemen umsetzen; die Bereitstellung einer Möglichkeit zur expliziten Modellierung der hier benötigten Applikationslogik grenzt dagegen die zur Verfügung stehenden Systeme stark auf neuere Ansätze ein. Portable Standards wie VRML97 oder Java3D kommen hier in Frage, sie lassen eine Modellierung der benötigten Funktionen (z.Bsp. über Fieldkonzepte) in Teilen zu, bieten aber nur beschränkte Echtzeitfähigkeit (je nach Renderer oder Interpreter) an und sind für das Szenario der hier angestrebten immersiven "Echtzeit"-VR-Anwendung als weniger geeignet anzusehen. Letztere stellt im Hinblick auf die einzusetzenden Verfahren so etwas wie eine modellierungstechnische Maximalanforderung und soll daher für die implementatorische Konzeptvalidierung eingesetzt werden.

Erste Prototypen sowohl für die benötigte Gestenerkennung auf Basis von PrOSA-Verfahren (s. Abschnitt 6.3.1) als auch für die angestrebte Erweiterung der Szenengraphobjekte und die darauf basierende Visualisierung wurden auf low-level Basis direkt in C/C++ implementiert. Für die Modellierung der graphischen Komponenten wurde direkt das Performer-Toolkit von SGI eingesetzt. Diese Prototypen dienten vor allem der Überprüfung der eingesetzten Methoden und der Offenlegung möglicher konzeptueller Schwachstellen im Design der multimodalen Eingabe. Auf dieser Basis wurden leicht modifizierte Anforderungen an die Kernkomponenten wiederholt spezifiziert und daraufhin etwaige Modellierungswerkzeuge evaluiert.

Unter Berücksichtigung der für die hier angestrebte Umsetzung gültigen Voraussetzungen, wurde auf eine Umsetzung mit Hilfe des AVANGO-Systems (s. Abschnitt 2.3) zurückgegriffen. In diesem werden sowohl Echtzeitbedingungen berücksichtigt[1] als auch Möglichkeiten der expliziten benötigten Datenflussmodellierung zwischen den Aktuatoren, Detektoren, Motion-Modifikatoren etc.gegeben.

Als Scriptsprache wird in AVANGO *Scheme*, eine auf LISP basierte funktionale Programmiersprache eingesetzt. Dieses ermöglicht es, viele zu parametrisierende Verfahren über ein Scriptinterface einem Rapid-Prototyping-Verfahren zugänglich zu machen und dabei variabel zu gestalten. Durch die Nähe zu Performer können alle berechnungsintensiven Verfahren direkt in einer maschinennahen Sprache (C oder C++) implementiert werden, die Parametrisierung und Zusammenschaltung geschieht dabei über eine High-level-Schnittstelle in Scheme. Darüber hinaus eröffnet sich so eine homogene Möglichkeit, sowohl die Sprachanalyse, das Parsing, als ebenso die Verfahren zur multimodalen Integration in einer gemeinsamen Umgebung umzusetzen. Die nahe Verwandschaft zu LISP ermöglicht den Einsatz vieler der im Rahmen der KI ausgearbeiteten Formalismen — z.Bsp. das ATN-Modell und seine hier entwickelte Erweiterung — in einer angemessenen funktionalen Darstellung.

Die von AVANGO dediziert zur Verfügung gestellte Lösung für die Datenkommunikation zwischen den entwickelten Komponenten, das bereits in Abschnitt 2.3 auf S. 42ff. vorgestellte Fieldcontainer-Konzept, ermöglicht hier die direkte Umsetzung der Bindungen zwischen Aktuatoren, Motion-Modifikatoren und Manipulatoren oder den Datenfluss der Attributsequenzen zum Referenzknoten oder zu und von den Detektoren. Diese Kommunikationsmöglichkeit wurde daher für die Umsetzung der Komponentenschnittstellen eingesetzt.

[1] Durch die generelle Benutzung des SGI Performer API.

6.3 Umsetzung der Konzepte im Prototyp

6.3.1 Ein externer Handformdetektor

Für die Erkennung von Formmerkmalen der Hände wurde ein Erkennermodul — ein Detektor — auf Basis des in Abschnitt 4.3 beschriebenen Template-matching implementiert. In diesem werden mögliche Formtemplates über Parametrisierung einfließender Gewichtungswerte und den im Template benötigten logischen Verknüpfungen in Grenzen definierbar gestaltet. Dieser Detektor dient vor allem der Überprüfung der Tragfähigkeit der gewählten Berechnungsverfahren und damit als Grundlage für die in AVANGO intergrierte modulare Version der PrOSA-Verfahren.

Mit Hilfe des Detektors werden die einzelnen Fingerstellungen gemäß ihres Streckungsgrades, der aktuellen Streckungsgeschwindigkeit und der Beschleunigung der Streckungsänderung bewertet. Diese drei Merkmale werden für jeden Finger berechnet und die entsprechenden Konfidenzen daraufhin je nach zu erkennender Geste gewichtet. Daraufhin folgt ein Test gegen Schwellwerte und anschließend eine logische Verknüpfung der Testresultate. Je nach Ausgang dieser Berechnungs- und Vergleichsschritte kann der Detektor auf die sechs verschiedenen, in Abbildung 4.13 auf S. 118 erläuterten Zustandsänderungen mit einer entsprechenden Nachrichtengenerierung reagieren. Er ist allerdings als ein externes Modul realisiert, da er keinen Zugriff auf geometrische Bezugssysteme benötigt. Der Detektor sendet daher nur bei den wichtigen Übergangsereignissen der Gestendetektion auf seiner rein internen Attributsequenz. Er ermöglicht folgende Operationen:

- Prädikatsberechnung für jeden Finger
- Dynamikauswertung durch erste und zweite Ableitung
- Vergleich mit Kalibrierungswerten
- Test gegen Schwellwerte
- Unterschiedliche Gewichtung nach Schwellwerttest
- Reaktion auf sechs verschiedenen Vergleichsmöglichkeiten

Mit Hilfe dieses Detektors wurden Templates zur Erkennung des Formanteils von Zeigegesten, dem Zugreifen und Öffnen der Hände und diverser symbolischer Gesten[2] gemäß der in Abschnitt 4.3 zum Teil bereits angegebenen

[2] Zum Beispiel zur Erkennung der verschiedenen mit der Hand darstellbaren Zahlsymbole.

Definitionen erstellt. Die einzelnen Berechnungsfaktoren wie Schwellwerte, Gewichte, in Grenzen die Art der logischen Verknüpfung und die Art der auf ein Berechnungsergebnis folgenden Reaktion sind als parametrisierte Werte dynamisch konfigurierbar. Abbildung 6.1 zeigt das graphische Frontend zur interaktiven Einstellung der verschiedenen Parameter bei gleichzeitigem Feedback über das Berechnungs- und Klassifikationsverhalten des Detektors.

Durch die externe Modellierung ergibt sich der Vorteil der Unabhängigkeit des Erkenners von anderen Prozessen und möglichen Flaschenhälsen durch aufwendige Berechnungsverfahren. Andererseits kann der Vorteil der Weitergabe von ganzen Attributsequenzen so nicht ausgenutzt werden. Abbildung 6.2 veranschaulicht einen ersten Ansatz mit einem Zusammenwirken verschiedener externer Komponenten, sowohl von Gestenerkennungs- als auch von CVK-spezifischen Agenten[3] im Kontext der VR-Anwendung. Durch diese externe Agentenmodellierung kann z.Bsp. eine Auslagerung von rechenintensiven Operationen auf physikalisch getrennte Systeme erfolgen. Die für die Gestenerkennung benötigten Kalibrierungswerte der verschiedenen auf Fingerbeugungswerten operierenden Detektoren werden dabei durch eine beliebige Instanz zur Verfügung gestellt und anschließend an alle Agenten mit gleicher Berechnungsgrundlage verschickt. Der Nachteil eines solchen Verfahrens ist die Einschränkung der Einsatzmöglichkeiten innerhalb einer VR-Anwendung durch den Verzicht auf die Attributsequenzweitergabe. Solange es sich um Matchingverfahren handelt, welche prinzipiell nur die Zustandsänderung des Erkennerresultats in einem diskreten Signal ermitteln, und solange die betrachteten Merkmale allein durch die sensorische — interne — Information berechnet werden können, ist eine VR-Anbindung unproblematisch. Dann werden vom Agenten nur kurze Signale übermittelt, welche in ausreichender Zeit in der VR-Anwendung dispatcht und bearbeitet werden können.

Unter den Gesichtspunkten der Einbindung der Aktuatoren und den damit verbundenen Attributsequenzen ist ein solcher extern-modularer Aufbau allerdings problematisch. Hier werden keine diskreten Zustandsänderungen mehr verschickt, sondern kontinuierlich Daten synchronisiert, Berechnungsergebnisse in eine gemeinsame Basis gebracht und weitergegeben. Ein Messagepassing-Prinzip müsste hier zum Datenstreaming mutieren. Zusätzlicher Overhead bei der Bearbeitung der so kontinuierlich und dazu asynchron ankommenden Daten ist für eine echtzeitfähige Anwendung als kritisch anzusehen. Weiterhin ist die Formklassifikation der Hände anhand der Fin-

[3]Hier findet das in der AG-WBS der Universität Bielefeld entwickelte Agentensystem *wbs-agents* Verwendung.

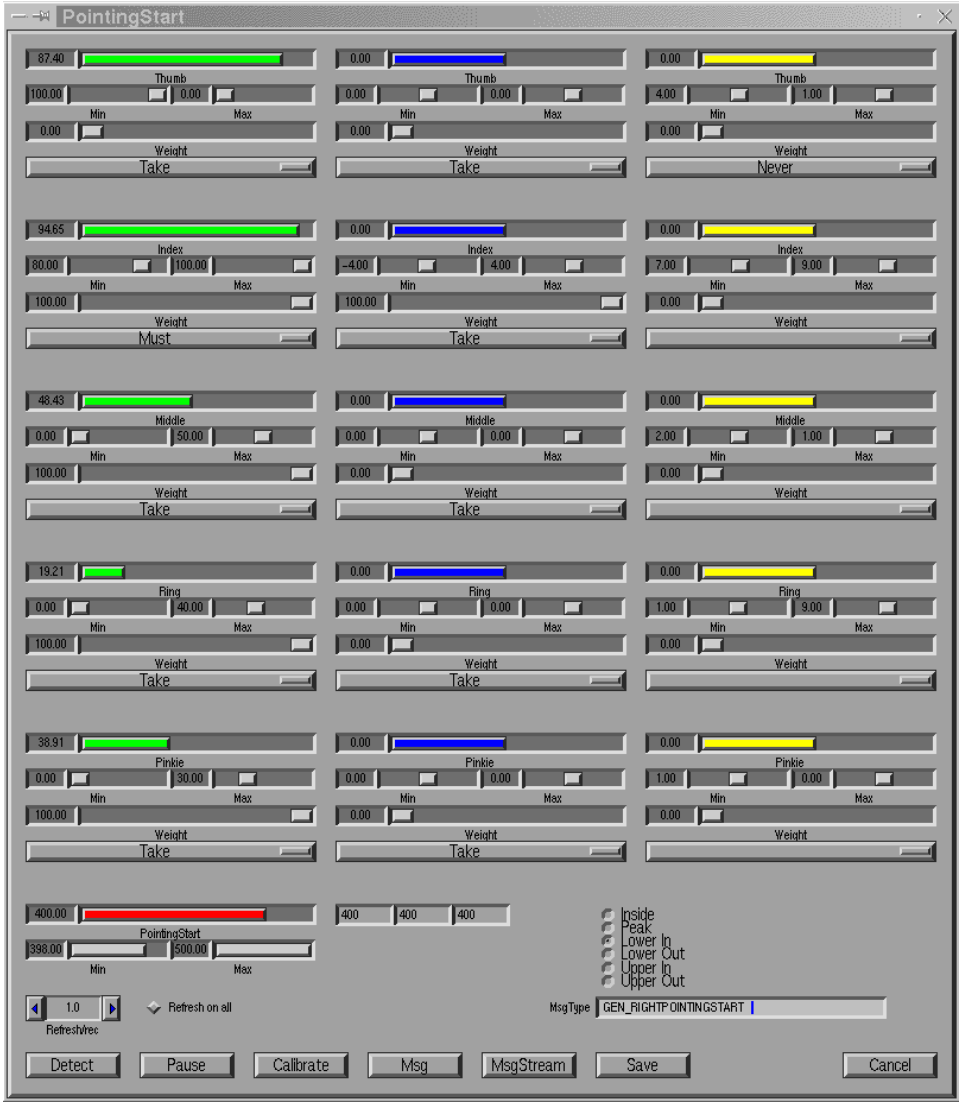

Abbildung 6.1: Graphisches Frontend des Handformdetektors mit Stellmöglichkeit für einzelne Parameter und Visualisierung des aktuellen Matching-Resultats.

gerbeugung nur ein Spezialfall der zu bearbeitenden Aufgaben. Viele der anderen Berechnungsvorgaben, zum Beispiel für die schritthaltende Auswertung der gestischen Deixis oder die Umsetzung mimetischer/kinemimischer Interaktion, basieren auf der kontinuierlichen Bereitstellung der Körperkon-

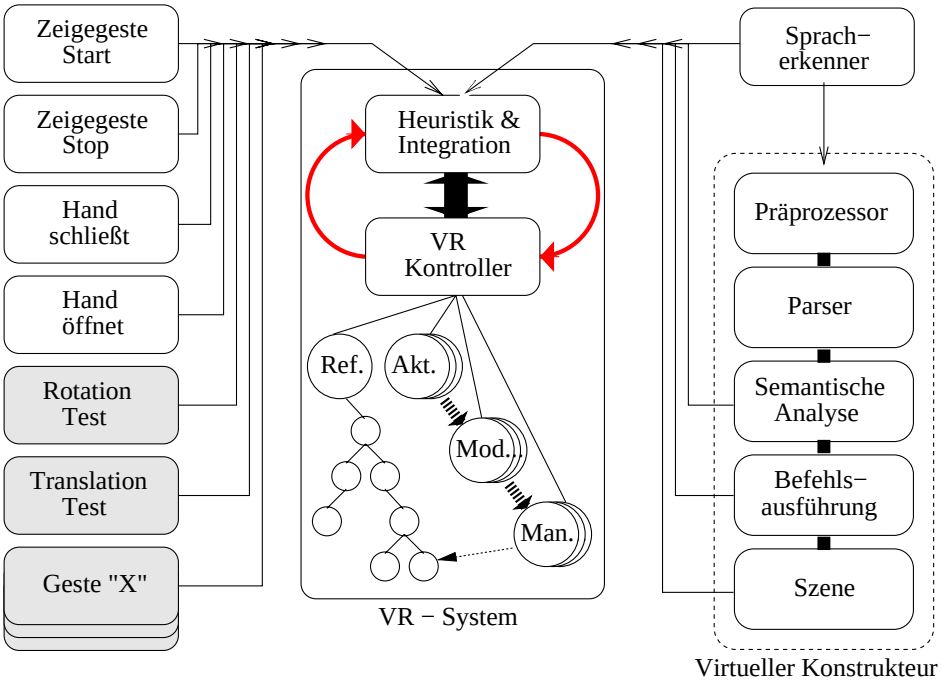

Abbildung 6.2: Externe eventgetriebene Gestendetektionsmodule sowie Anbindung der Spracherkennerkomponente und der CVK-Agenten. Die externe Modellierung der Detektoren führt zu Schwierigkeiten bei der kontinuierlichen Auswertung der Bewegungsinformationen (s. Text).

figurationsdaten in einer einheitlichen Basis. Aus diesem Grund wurden ja die Konzepte der Aktuatoren, Motion-Modifikatoren und Manipulatoren entwickelt. Hier werden die notwendigen Berechnungen innerhalb der App-Phase der eigentlichen Render-Loop verankert.

6.3.2 Interne PrOSA-Detektoren

Um die im vorigen Abschnitt erörterten Schwierigkeiten zu lösen, fand im Rahmen des SGIM-Prototyps eine Implementierung besonders der für kontinuierliche Auswertung benötigten Komponenten mit einer direkten Einbettung in den VR-Ablauf statt (s. die grau schattierten Objekte in Abbildung 6.3). Die dermaßen implementierten Detektoren erhalten nun aktuelle Daten der Körperkonfiguration über die Attributsequenzen von den jeweiligen Aktuatoren. Um die Ergebnisse der Berechnungen anderen Modulen zur Verfügung zu stellen, sind diese Detektoren mit einer Scheme-

Schnittstelle versehen. Sie erlauben so entweder eine direkte Kopplung an die Auswertungsergebnisse über die Fieldkonzepte oder eine Kombination verschiedener Erkenner über die zur Verfügung gestellten Attributsequenzen. Zu den bisher umgesetzten Verfahren zählen, neben den im prototypischen Formerkenner evaluierten Methoden, nun ebenfalls solche zur genaueren Analyse von beschriebenen Trajektorienbewegungen, um so z.Bsp. die mimetisch/kinemimische Interaktion zu gewährleisten. Darüber hinaus wurden bereits Vorkehrungen für eine Entwicklung von Detektoren zur Erkennung linearer Bewegungen und Parallelitäten getroffen.

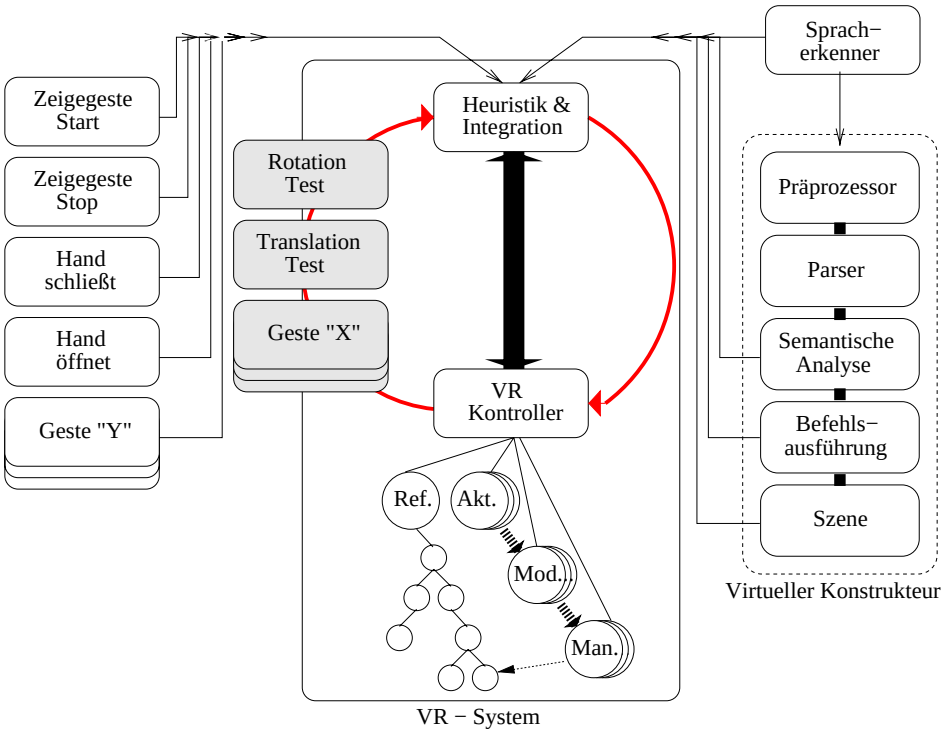

Abbildung 6.3: Einbettung eng gekoppelter vs. eventgetriebener Gestendetektionsmodule in den VR-Ablauf. Erst die Integration der Detektorfunktion in die VR-Loop gewährleistet auch eine reibungslose kontinuierliche Auswertung.

6.3.3 Implementation der VR-Konzepte

Die Basiskonzepte Aktuator, Motion-Modifikator und Manipulator sind die Hauptkomponenten einer Einbettung der Gestenerkennung und der multimodalen Integration in den VR-Kontext. Ihre Umsetzung erfolgte für benötigte Beispielkomponenten komplett mit Hilfe des AVANGO-Toolkits. Die entwickelten NDOF-Bewegungsaktuatoren für beide Hände und den Kopf wurden dabei mit einer zusätzlichen Kalibrationsmöglichkeit versehen, um Unzulänglichkeiten in den sensorischen Befestigungen auszugleichen. Über ein graphisches Feedback können die Richtungen in den Aktuatoren verankerter Referenzstrahlen online inspiziert sowie *sprachlich* verändert und so den aktuellen Gegebenheiten angepasst werden.

Diese Aktuatoren liefern über die Fieldverbindungen die Informationen der primären Referenzstrahlen an einen Referenzknoten. Die aktuell implementierte Instanz dieses Konzepts verwaltet alle sprach-gestisch referenzierbaren Objekte sowohl in Bezug auf ihre Position in der Szene, um die Auswertung des deiktischen mappings zu ermöglichen, als auch in Bezug auf ihre Attribute. Von letzteren sind im Prototypen Farben- und Typenbezeichnungen implementiert und somit einer sprachlichen Benennung zugänglich gemacht worden. Der gewählte Ansatz zur Referenzauflösung bedient sich der bereits erläuterten Abstandsberechnungen und Mengenoperationen (s. Abschnitt 5.4.2). Die Blickrichtung während der sprachlichen Äußerung liefert die in Frage kommenden Basisobjekte, sprachliche Attributbezeichnungen schränken diese ein. Wird keine offensichtliche Gestik benutzt, so definiert allein die vorgegebene Sortierung der Spacemap, gegeben durch die Blickrichtung, das in Frage kommende Objekt. Bei einer Erkennung einer oder mehrerer Zeigegesten entscheidet die multimodale Integration gemäß der angegebenen Auswertefunktion der Zeigegesten den Aufbau und die Objektordnung in der resultierenden Objekmenge.

Manipulatoren, welche ein direktes Greifen, ein Verschieben, ein Rotieren und ein Lösen ausgewählter Objekte zur Verfügung stellen, sind in einer ersten Realisation für kontinuierliche Manipulationen erstellt worden. Ein vollständig gestaffeltes System mit Aktuator, Motion-Modifikator und Manipulator wurde für eine Bereitstellung einer Objektlageveränderung mit Hilfe eines Vormachens der gewünschten Rotation implementiert. Verbindungen lassen sich auf zwei unterschiedliche Interaktionsarten etablieren: Sie sind in einem diskreten Schritt umzusetzen, indem Äußerungen der Form „ ... *nimm* [Zeigegeste] *dieses gelbe Rohr und verbinde es mit* [Zeigegeste] *dem* ... “ Verwendung finden. Durch den Einsatz einer echzeitfähigen Kollisionskontrolle können Objektlagen und -positionen aber auch kontinuierlich manipuliert

werden. Kommt es dabei zu einem Kontaktschluss und ist ein Verbindungswunsch — über einen speziellen Anwendungsmodus — aktiv, so werden die kollidierten Objekte auf Verbindungsmöglichkeiten (Ports) an den Kollisionsstellen getestet und daraufhin die Objektfügung ggfs. etabliert.

Letztlich wurde als zentrale Komponente eine Version des erweiterten ATN sowie die im Rahmen der Anwendungslogik benötigte Funktionalität auf der Scheme-Ebene implementiert. Besonders bei der Verankerung verschiedener Funktionen an den ATN-Kanten — von den benötigten Tests der sprach-gestischen Perzepte bis hin zu den Auswertefunktionen — zeigt sich die Stärke der Scheme-eigenen gemeinsamen Repräsentation von Daten und Funktionen. Ohne allzugroßen Aufwand lässt sich so das Zustandsmodell samt verbindenden Kanten mit den dazu gehörigen Funktionen aufbauen sowie der Auswertealgorithmus umsetzen. Da die zur Auswertung benötigten Komponenten so ebenfalls eingebunden werden (auch sie haben eine Scheme-Schnittstelle), ergibt sich hier eine einheitliche Verbindungs- und Integrationsebene.

6.4 Beispielinteraktionen

An Hand von illustrierten Beispielen werden in den folgenden Abschnitten Interaktionen im und mit dem SGIM-System erläutert. Diese basieren ausschließlich auf den bisher ausgeführten Konzepten und der dargelegten Systemmodellierung und sollen so einen Einblick in die Möglichkeiten der multimodalen Interaktion in einer Virtuellen Umgebung vermitteln. Neben der Implementation der neuen VR-Knotenkonzepte für sprach-gestische Interaktion, wurden ebenfalls rein sprachliche Möglichkeiten z.Bsp. zur Objekterzeugung, zur Steuerung spezieller Animationssequenzen oder zur Navigation zu ausgezeichneten Orten in der virtuellen Szene geschaffen. Die folgenden Abbildungen sollen einen Eindruck der bereits im praktischen Einsatz erprobten und Verwendung findenden Verfahren vermitteln. Um eine bessere Druckqualität der Bilder zu gewährleisten, wurde — nur für die Bildaufnahme — auf die sonst übliche stereoskopische Darstellung mit Hilfe zweier versetzter Halbbilder verzichtet. Etwaige Verzerrungen sind auf die eingesetzte benutzerspezifische Perspektivkorrektur zurückzuführen. Die jeweiligen stattfindenden Interaktionen sind in den Abbildungsunterschriften ausgeführt.

6.4.1 Das virtuelle Labor

Abbildung 6.4: Ein Anwender betritt das virtuelle Labor. Rein sprachliche Anweisungen („ ... *öffne das Schott!*“) lösen spezielle Aktionen — hier das Öffnen einer Tür — in der Szene aus.

Abbildung 6.5: Ein Blick in den Eingangsbereich des virtuellen Labors. In diesem Umfeld finden die Interaktionen mit dem SGIM-System statt.

Abbildung 6.6: Ein Beispiel für die Navigation in der Szene: Verschiedene Orte des Labors können durch Benutzeräußerungen der Form „ ... *geh zu Spot* X!“ angesteuert werden. Das letzte Bild der Sequenz zeigt die Ankunft des Anwenders am Konstruktionsort mit vier dort bereits vorhandenen Bauteilen.

6.4.2 Objekterzeugung

Abbildung 6.7: Ein neues Objekt wird vom Anwender durch eine rein verbale Äußerung („ ... *gib mir ein rotes Rad* ... “) in die Szene eingefügt.

6.4.3 Selektion von Objekten

Abbildung 6.8: Ein Beispiel für die sprach-gestische Auswahl (Selektion) eines gewünschten Objekts. Die gemeinsame Auswertung a) der aus VP und DNP bestehenden (Teil-)Äußerung („ ... *nimm dieses Rohr* ... “) mit b) der Zeigegeste und c) der aktuellen Blickrichtung, führt im Beispiel zu einer Hervorhebung des entsprechenden Bauteils, eines länglichen Vierkantrohrs.

6.4.4 Verschieben

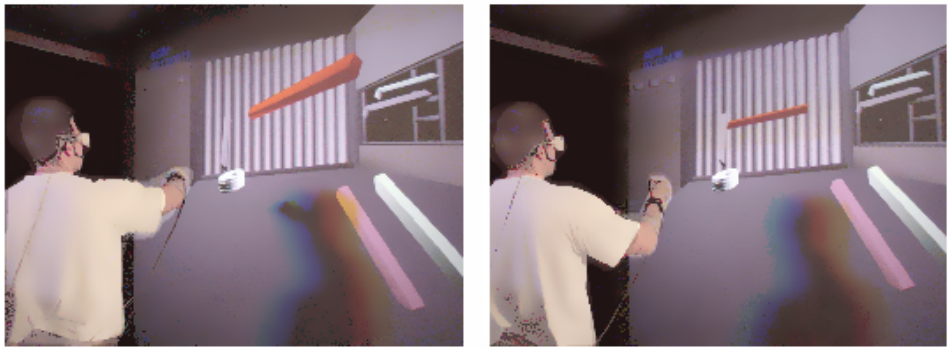

Abbildung 6.9: Nach erfolgter Selektion vermittelt eine Greifgeste eine Bindung zwischen entsprechender Hand und dem Objekt. Darauf folgende Handbewegungen, werden als analoge Lage- und Positionsänderungen des Objekts interpretiert — ein distantes dragging: In den Bildern wird das Vierkantrohr nach links verschoben und dabei etwas gedreht.

6.4.5 Rotieren

Abbildung 6.10: Eine kontinuierliche Interaktion — in diesem Fall eine Rotation — wurde durch die verbale Äußerung „ ... *dreh das Rad so herum* ... “ angestoßen und anschließend durch die mimetisch/kinemimische Geste vorgemacht und umgesetzt.

6.4.6 Verbindungen diskret etablieren

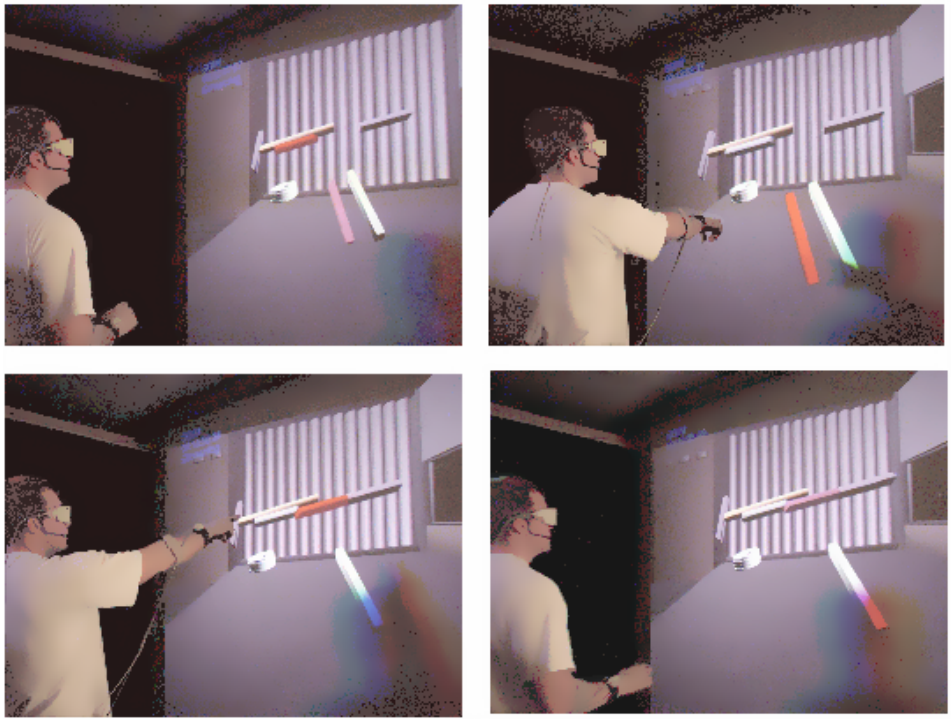

Abbildung 6.11: Eine Verbindung wird diskret etabliert. Dabei wurden von oben links nach unten rechts folgende Äußerungen verwendet: „ ... *nimm das* [Zeigegeste] *violette Vierkantrohr und verbinde es mit* [Zeigegeste] *dem weißen Teil!*“

6.4.7 Verbindungen kontinuierlich etablieren

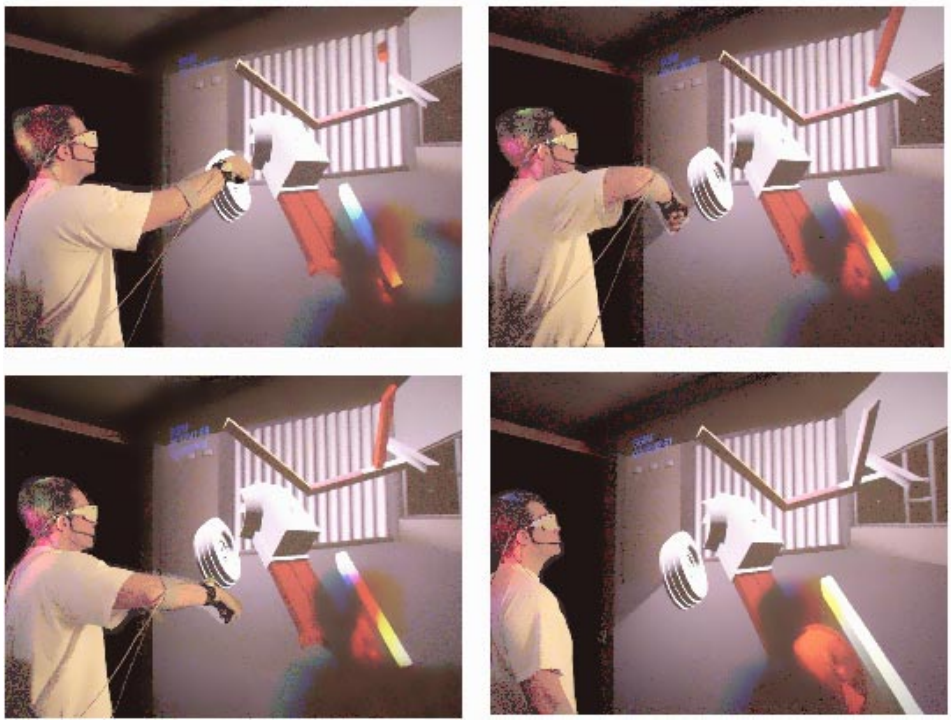

Abbildung 6.12: Eine Verbindung wird kontinuierlich etabliert. Der Benutzer führt ein neues Bauteil von oben auf das bereits zusammengesetzte Teilaggregat. Nachdem eine Kollision zwischen den Objekten erkannt wurde, hat eine Anfrage an das wissenbasierte System zwei mögliche Verbindungsorte (Ports) in der Nähe der Kollisionsstellen zurückgeliefert. Die entsprechenden Teile werden gemäß dieser Information gegeneinander ausgerichtet und *schnappen* zusammen.

6.5 Zusammenfassung

Das SGIM-System repräsentiert eine prototypische Implementation der in dieser Arbeit vorgestellten Konzepte. Während deren Umsetzung konnten im Verlauf weitere nützliche Informationen über das Systemdesign gewonnen werden. Die Alternativen mit — aus Sicht der Render-Loop — externen oder eingebetteten Detektoren hat gezeigt, dass letztere Version

in Bezug auf Wartbarkeit und Performanz der externen Variante deutlich überlegen ist. Eine konsequente Verfolgung der in den Szenengraphen gekoppelten multimodalen Schnittstelle erlaubt überdies eine Portabilität auf andere VR-Anwendungen. Dabei konnten die erarbeiteten Komponenten immer weiter granularisiert werden, so dass die Bereitstellung einer multimodalen Schnittstelle nun durch die Erstellung eines *Toolkits* von PrOSA Komponenten unterstützt wird. Eine Adaption auf ein anderes bestehendes oder geplantes Szenengraph-API[4] setzt hier als Minimalforderung eine zur herkömmlichen Graphenstruktur orthogonale Verbindungs- bzw. Kommunikationsschnittstelle (etwa durch ein Fieldkonzept) für den Informationsfluss voraus.

Die vorgestellten etablierten Interaktionsformen verdeutlichen die Stärken eines multimodalen Zugangs zu virtuellen Welten. Sprache allein erlaubt das einfache Auslösen von Aktionen ohne eine Zuhilfenahme dedizierter Eingabegeräte. Im Zusammenspiel mit Gestik lassen sich dabei 1) Amiguitäten — etwa bei der Referenzermittlung — erfolgreich vermeiden bzw. auflösen und 2) Aktionen konkret auch in ihren Parametern spezifizieren. Eine gestische Kommunikation erweist sich hier als besonders nützlich, da sprachliche Konstrukte für einen solchen Zweck unzureichend und unnatürlich erscheinen. Mimetische/kinemimische Gesten erlauben sogar die schrittweise Verfolgung der Interaktionsumsetzung, also die Beibehaltung der Vorteile eines interaktiven Graphiksystems. Selbst die sonst in vielen VR-Szenarien als problematisch angesehene Navigation lässt sich — wie gezeigt — sprach-gestisch realisieren. Eine objektive Aussage zur Nützlichkeit der hier vorgestellten Form einer Schnittstelle setzt natürlich noch zu erbringende weitergehende Untersuchungen voraus. Subjektiv bewertet kann dagegen bestätigt werden, dass ein multimodales Interface im Vergleich zu WIMP-ähnlichen Schnittstellen Interaktionen in weiten Teilen adäquat, und in vielen Bereichen deutlich intuitiver bereitstellen kann.

[4] Application Programming Interface

Kapitel 7

Resümee

Gegenstand dieser Arbeit ist die Exploration von Möglichkeiten zur multimodalen, sprach-gestischen Interaktion mit Virtuellen Umgebungen anhand eines konkreten Beispiels aus der Domäne der Virtuellen Konstruktion. Zu diesem Zweck wurden die angesprochenen Forschungsteilgebiete (sowohl der Informatik als auch zum Teil der Psycholiguistik und Gestenforschung) unter einem gemeinsamen Blickpunkt beleuchtet: Dem Ziel der Offenlegung der Zusammenhänge zwischen Sprache und Gestik in einer instruktionsgeleiteten Interaktion und deren Modellierung in Konzepten, welche eine spätere konkrete technische Realisation ermöglichen. Die Betrachtungen bisher Verwendung findender VR-Eingabeverfahren haben deren Unzulänglichkeiten — besonders bei einem Einsatz immersiver Displays — verdeutlicht und eine Form der multimodalen Interaktion für solche Anwendungen motiviert. Der Anspruch des Einsatzes in VR-Anwendungen stellt dabei auf der einen Seite hohe Anforderungen an die methodische Einbindung — von unterschiedlich synchronisierten Datenquellen und Ablaufstrukturen bis zur Bereitstellung einer funktionalen Gestenerkennung und multimodalen Integration, bietet auf der anderen Seite aber die Chance, Gestik als spatial ausgeprägte menschliche Artikulation direkt in einem den Auswertungsberechnungen zugänglichen Raumreferenzsystem zu verankern und somit eine der Grundlagen für eine natürliche sprach-gestische Interaktion in der VR zu gewährleisten. Dafür wurden unter Berücksichtigung bestehender VR-Modellierungsmethoden weitgehend allgemeingültige Verfahren zum Einsatz von multimodalen Schnittstellen in der VR ausgearbeitet. Die erarbeiteten Konzepte wurden anschließend mittels einer exemplarischen technischen Umsetzung im SGIM-System erprobt.

7.1 Ergebnisse

Die Betrachtungen des Forschungsstands zur Gestik, zu ihrem kommunikativen Gebrauch, sowie ihrer spatiotemporalen Ausprägung lassen sich folgendermaßen zusammenfassen: Gestik ist besonders in solchen Fällen als Kommunikationsmittel geeignet, wenn Information mit direkt oder metaphorisch spatialem Gehalt übermittelt werden soll oder verbale Informationen unzureichend sind. Dies gilt insbesondere im Kontext der herausgearbeiteten Interaktionsgestik: Dem Gebrauch coverbaler deiktischer, mimetischer/kinemimischer und ikonischer (spatiographischer und pictomimischer) Gesten. Deixis steht in engem Zusammenhang mit der Kommunikation von Referenten im Diskurs. Die gestische Deixis schränkt hier den für die Referenzvermittlung in Frage kommenden Raum eines Rezipienten ein. Mimetische/kinemimische und ikonische Gestik dienen von einem direkten kommunikativen Gesichtspunkt aus unterschiedlichen Zwecken: Sie drücken entweder statische Eigenschaften oder dynamische Veränderungen aus. Anhand ihrer spatiotemporalen Ausprägung lassen sich diese aber nicht unterscheiden. Daraus resultiert, dass ihre endgültige Interpretation nur im sprachlichen Kontext erfolgen kann. Ein rein merkmalsbasierter Ansatz ist hier, im Gegensatz zu der Aufgabe einer Interpretation symbolischer Gesten, unzureichend. Letztere haben eine definite Form und eine direkte semantische Abbildung, sind aber erlernt und damit weniger natürlich als die zuvor angesprochenen Gestentypen.

Die Analyse möglicher Gestenerkennungsmethoden hat rein symbolische Kodierungen auf bestehenden Notationssystemen als unzureichend für die hier angestrebte Interaktionsumsetzung bewertet. Der spatiotemporale Verlauf ikonischer und mimetischer/kinemimischer Gestik drückt explizit quantitative Aspekte übermittelter Information aus. Zum Beispiel sind Informationen über den Verlauf in einer bestimmten Ebene, die Lage dieser Ebene, die Geschwindigkeit der Bewegung oder andere mit der beschriebenen Trajektorie direkt verknüpfte Bahneigenschaften wichtige Daten bei der Analyse eines gestischen *Vormachens* von Form oder Bewegung. Daher wurde zwar bei der Erkennung der Gestik durch die Einführung der Detektoren zum Gestentemplatematching und den damit verbundenen Bewertungsprädikaten auf eine symbolische Form, bei deren Auswertung aber explizit auf nummerische Analysedaten zurückgegriffen, also eine hybride Repräsentation gewählt. Die darauf basierende Entwicklung der PrOSA-Verfahren berücksichtigt diesen Umstand durch die Einführung der Attributsequenzen, welche Modellierungen auf beiden Repräsentationsseiten zulassen. Um die Gestenerkennungsaufgabe zu lösen, wurden acht verschiedene spatiotempo-

rale Gestenmerkmale vorgeschlagen und für eine Auswahl dieser die zugrundeliegenden Bewegungsdaten und deren analytische Berechnungsverfahren in den Aktuatoren (s.u.) verankert und als Attributsequenzen weiteren Berechnungsschritten zur Verfügung gestellt. Auf der Basis dieser Informationen wurden die Auswerteprädikate für die in der Interaktionsauswertung benötigten Gestenformen bestimmt.

Die Betrachtungen bisheriger Interaktions- und Modellierungmethoden in der VR führten zu der Entwicklung von allgemeinen Konzepten einer Anbindung sprach-gestischer Interaktionen im Kontext von VR-Anwendungen. In einem ersten Schritt erfolgte eine Herausarbeitung der in der VR benötigten Interaktionen und ein Abgleich dieser mit Möglichkeiten einer multimodalen Realisierung. Dabei konnten zwei grundsätzlich unterschiedliche Interaktionsformen — diskret und kontinuierlich — unterschieden werden, welche sich durch die Form der Umsetzung ausprägen.

Unter Berücksichtigung typischer in der VR eingesetzter Anwendungs- und Graphstrukturen wurde mit der Einführung von Aktuatoren, Motion-Modifikatoren, Manipulatoren, Detektoren sowie Referenzknoten die Möglichkeit geschaffen, bisherige Eingabemethoden konzeptuell um multimodale Interaktionsfähigkeiten zu erweitern. Die entwickelten Modelle sind hier in weiten Maßen unabhängig von einer konkreten VR-Implementierung; sie sind in Übereinstimmung mit den Ergebnissen aus Untersuchungen der Gemeinsamkeiten neuartiger VR-Modellierungswerkzeuge entwickelt worden. Dieses Vorgehen orientierte sich an den aktuellen Bestrebungen, eine weitgehende Unabhängigkeit von konkret eingesetzten VR-Systemen, Ausgabekonfigurationen, Displayarten u.Ä.modular zu unterstützen und erweitert dieses Vorgehen um einen Vorschlag einer allgemeinen multimodalen Interaktionseinbettung in die VR.

Bei der konzeptuellen Entwicklung eines Systems zur multimodalen Interaktion in der VR lassen sich folgende Ergebnisse dokumentieren: Unter Berücksichtigung der Forschungsergebnisse sowie anhand der Untersuchung von Beispieläußerungen der gewählten Domäne konnten drei Formen von sprach-gestischen Zusammenhängen für die Aufgabe einer Auswertung multimodaler Äußerungen explizit gemacht werden: Temporale, semantische (wenn nicht durch temporalen Zusammenhang gegebene) und kontextabhängige Äußerungseigenschaften auf verschiedenen Ebenen ermöglichen eine gelingende Zusammenführung von Gestik und Sprache für das Zielszenario. Für die deiktischen Gesten findet sich ein temporaler Bezug vor allem zu den einzelnen Konstituenten in den DNPs — hauptsächlich zu den Demonstrativpronomen, Artikeln, Lokalattributen und Lokaladverbien. Als zu ikonischen und mimetisch/kinemimischen Gesten korrespondierende sprach-

liche Einheiten wurden einerseits Modaladverbien aber andererseits auch der Typus eines entsprechend geäußerten Prädikats oder evtl. geäußerte Formattributadjektive innerhalb der DNPs identifiziert.

Um die Integrations- und Interpretationsaufgabe auf Basis der oben genannten Beziehungen für die multimodalen Eingaben zu realisieren, wurde das erweitere ATN-Modell konzipiert. Dieses reichert den ursprünglichen ATN-Formalismus um die zeitliche Repräsentation des Traversierungsvorgangs an und ermöglicht so die Auswertung zeitlich verorteter Eingaben aus unterschiedlichen Quellen. Darüber hinaus wurde der ursprünglich rein unimodal eventgetriebene ATN-Traversierungsalgorithmus sowohl für den Einsatz in einer geschlossenen Simulationsschleife — der Render-Loop — als auch für die Bearbeitung mehrerer zugrunde liegender Informationsquellen adaptiert.

Für die Korrelation zwischen DNP auf linguistischer Ebene und gestischer Deixis wurde ein Dereferenzierungsverfahren entworfen, welches auf Mengenoperationen basiert. Dieses betrachtet sowohl verbale Information als auch die durch die Blickrichtung und Armbewegungen definierte Körperdeixis in einer gemeinsamen Repräsentation. Im hier verwendeten Modell gestischer Deixis werden Referenzsstrahlen bestimmt, welche durch eine eindeutige Richtung bestimmter Körperteile den umgebenden Raum segmentieren und die darin befindlichen Objekte inhärent ordnen. Das dafür verwendete Modell ist dabei in hohem Maße — auch kontinuierlich — parametrisierbar und liefert mit der hier bereits erarbeiteten Heuristik sehr gute Ergebnisse bei der sprach-gestischen Referenzvermittlung im exemplarischen System.

Um die erarbeiteten Konzepte zu überprüfen, wurde eine exemplarische Implementation einer Interface-Komponente für eine Anwendung der Virtuellen Konstruktion vorgenommen. Die Grundformen multimodaler Interaktion lassen sich an diesem Prototypen erproben. Als besonders geeignet erweist sich hier die natürliche multimodale Referenzierung der Objekte. Das gewählte Integrationsmodell für die Deixis arbeitet zuverlässig und ermittelt selbst bei vielen — evtl. auch sehr ähnlichen — Objekten die gewünschten Referenten schnell. Die damit gegebene Unabhängigkeit von der Distanz zu den Objekten bietet hier die für die VR sehr wichtige Erweiterung des Interaktionsraums. Daneben können im Prototypen ebenso Aktionen ausgelöst sowie mimetisch/kinemimische Interaktionen (am Beispiel der Rotation) durchgeführt werden. Die Möglichkeit zur sprachlichen Objektinstanziierung sowie zwei verschiedene Formen der Verbindungserstellung setzen bereits die für die VK benötigte Grundfunktionalität um.

7.2 Diskussion

Die Frage, ob multimodale Interaktion in Virtuellen Umgebungen adäquat ist und eine Alternative zu den herkömmlichen Methoden bietet, lässt sich objektiv nur über quantitative vergleichende Studien zwischen beiden Vorgehen beantworten (s. Abschnitt 6.5). Die hier dazu getroffenen Aussagen beruhen auf den subjektiven Erfahrungen sowohl mit dem Umgang der bisher eingesetzten als auch mit der Erprobung der neuen Interaktionsmethoden. Besonders die Auswahl von Objekten, das Referenzieren und Selektieren erweist sich in der VR, gegeben durch die zusätzliche dritte Dimension, als eine mit den point-and-click Metaphern teilweise sehr schwer umzusetzende Aufgabe. Die in bisherigen Vorhaben (s. Abschnitt 2.1) realisierten Selektionsmöglichkeiten durch Zeigen und Deixis weisen unterschiedliche Schwachstellen auf, zu deren Lösung der Vorschlag der hier erbrachten Konzeption einer sprach-gestischen Dereferenzierung gemacht wurde. Die entwickelte Methode verzichtet vollständig auf zusätzliche Hilfsmittel (z.Bsp. sichtbare Cursor) und ermöglicht ein natürliches, durch etwaige feste Dialogstrukturen unrestriktiertes multimodales Äußerungsverhalten eines Anwenders. Dass eine *sprachliche* Eingabe die Auswahl bestimmter Werkzeuge und Anwendungsmodi in der VR beschleunigt, wurde dagegen bereits durch Studien belegt. Für die ausgearbeiteten Basisinteraktionen wird bestätigt, dass die zusätzliche gestische Modalität als ein Ersatz für bestimmte Werkzeuge dienen kann.

Die ausgewählte Methode einer Gestenanalyse und -erkennung regt weiterhin zur Diskussion an. Ob eine rein symbolische Notation als Grundlage einer Repräsentation gestischer Artikulation ausreichend ist, kann für den geschilderten Interaktionseinsatz verneint werden. Dafür ist die Granularitätsstufe existierender Notationen zu grob. Sie kodieren Gestik auf einem Niveau, auf welchem eine kontinuierlich schritthaltende oder auch später erfolgende genaue Rekonstruktion der Trajektorie im Hinblick auf eine kontinuierliche Interaktion nicht durchzuführen ist. Daher wurde in dem hier entwickelten Ansatz eine hybride Repräsentation eingesetzt, welche sich insbesondere im Zusammenhang mit den kontinuierlichen Interaktionen als adäquat bestätigt hat.

Die Form der entwickelten und in den VR-Ablauf eingebetteten Konzepte bietet eine weitgehend allgemeine Modellierungsmethode zur Erweiterung bestehender VR-Anwendungen. Die dabei eingesetzten Fieldcontainer zur Kommunikation zwischen den Komponenten erweisen sich zwar einerseits als leicht portabel und adaptierbar, können aber andererseits durch die mit ihrem Arbeitsprinzip verknüpften Datenkonvertierungen und Über-

tragungswegen evtl. schnell an ihre Leistungsgrenze kommen. Dieses gilt insbesondere für die Weitergabe der Attributsequenzen. Hier müssen ggfs. weitere alternative Verfahren eingesetzt werden. Dagegen ermöglicht die Strukturierung der Teilaufgaben einer multimodalen Eingabe in der getroffenen Wahl ausgearbeiteter Komponentenklassen (Aktuatoren, Detektoren, Motion-Modifikatoren, ...) eine geeignete Modularisierung in für sich gekapselte Aufgabenfelder. Diese berücksichtigen dabei ebenfalls schon existierende Verfahren (s. Manipulatoren) und führen eine geeignete Abstraktionsschicht zwischen Eingabegeräten und deren Auswertung innerhalb einer Anwendung ein.

Weiterhin stellen die bisher eingesetzten Verfahren zur Analyse der beschriebenen Trajektorie während einer gestischen Artikulation nur eine von vielen Möglichkeiten dar. Das Problem, welches es hier zu bewältigen gilt, ist einen geeigneten Kompromiss zwischen Latenzzeit und Verlässlichkeit der Analyse zu finden. Die im Prototypen verankerten Methoden konzentrieren sich vor allen Dingen auf die geringe Latenzzeit, reagieren dadurch aber noch recht sensibel auf Fehler, wie sie beispielsweise durch die eingesetzte elektromagnetische Sensorik hervorgerufen werden können. Zusätzliche Filterstufen oder Interpolationen über Spline-Verfahren stellen für diesen Einsatz gute Alternativen zur Verfügung und sollen weiter exploriert werden.

Der ausgearbeitete und eingesetzte erweiterte ATN-Formalismus ermöglicht die explizite Darstellung der multimodalen Zusammenhänge in einer gemeinsamen Repräsentation. Das ATN-Modell selber lässt bereits sehr viele Freiheiten der zu modellierenden Domäne zu. Der Nachteil liegt in der Voraussetzung, dass damit wirklich alle möglichen Varianten von multimodalen Zusammenhängen im Vorfeld in jedem Detail definiert werden müssen. Die Stärke dieser Methode — die gemeinsame Verabeitung von syntaktischen, semantischen und pragmatischen Informationen — kann sich bei einer gewünschten Erweiterung der Domäne schnell in einen Nachteil verwandeln. Es existieren keine Verfahren einer Abstraktion über verwendete ähnliche Integrationsregeln. Für den eingesetzten Fall einer begrenzten Wortschatzmenge und den damit möglichen sprachlichen Konstrukten sowie den klar eingegrenzten Interaktionstypen fällt dieses aber noch nicht ins Gewicht.

7.3 Ausblick

"*It has just begun*" — mit den vorgelegten Ergebnissen lassen sich einerseits bereits interessante Aussagen über die Nützlichkeit multimodaler Interaktion in der VR treffen, andererseits konnte eine der Stärken der beschriebenen

Verfahren — die Möglichkeit der Übertragung auf andere Domänen und VR-Systeme durch weitgehend allgemeingültige Konzepte — erst in exemplarischen Umsetzungen für zwei Arten von VR-Umgebungen herausgestellt und ausgenutzt werden.

Auch sind die Möglichkeiten insbesondere der gestischen Interaktion bei weitem nicht ausgeschöpft. Konzeptionell bereits vorgesehen ist die Interpretation ikonischer Gestik und die Formbeschreibung von Objekten als Erweiterung der Dereferenzierung durch multimodale Eingaben. Als Grundlage dafür werden im Rahmen weiterer Arbeiten automatische Verfahren entwickelt, um beliebige graphische Objekte[1] auf bezeichnende Formattribute durch die sie konstituierenden begrenzenden Flächen, Kanten etc.hin zu untersuchen und diese Informationen der multimodalen Dereferenzierung zugänglich zu machen. So kann neben Typ- und Farbattributen die Beschreibung der Objektform — sprachlich und gestisch — in den Prozess der Dereferenzierung aufgenommen werden. Die hier entwickelten Komponenten sind für solche Erweiterungsmöglichkeiten bereits vorbereitet. Daran anschließend sollte ein Konzept ausgearbeitet werden, um benötigte Objektattribute entweder in den einzelnen Objektmodellen als Erweiterung zu verankern oder, wie beschrieben, durch automatische Verfahren zum Instanziierungszeitpunkt zu generieren. Damit könnte eine portable Lösung einer Bereitstellung von Attributwissen über Objekte für eine multimodale Schnittstelle geschaffen werden.

Möglichkeiten von Erweiterungen in Bezug auf die Referenzanalyse sind damit aber noch lange nicht erschöpft. Die eingesetzte Abbildungsfunktion zur Erstellung der Spacemapsortierung ist in der exemplarischen Umsetzung eine erste Näherung einer von vielen möglichen Ordungsfunktionen. Vor allen Dingen wurde damit die Methode einer konzeptuellen Einbettung der Funktionalität geschaffen. Obwohl bereits die eingesetzten Heuristiken sehr gute Ergebnisse liefern, sollte es ein Bestreben sein, diese Heuristiken beispielsweise auf Ergebnisse aus Experimenten zur Zeigegenauigkeit oder zur Betrachtung der Blickrichtungssignifikanz bei der Referenzierung zu modifizieren. Die Art der verwendeten Abstandsberechnung ist dahingehend parametrisiert und lässt sich leicht adaptieren sowie auf konkrete experimentelle Untersuchungsergebnisse einstellen.

Aus dem Bereich der möglichen Interaktionen wurden im Prototypen bereits einige Methoden realisiert. Doch existieren neben den beschriebenen viele weitere. Ikonische Gestik wird zum Beispiel auch zur Verdeutlichung einer gewünschten Veränderung benutzt ("*Mach es* [Ikone] *so groß*"). Diese

[1] Wie sie mit herkömmlichen Modellierungswerkzeugen erstellt werden können.

Manipulationsart wurde bereits als eine der Grundmanipulationen in der VR identifiziert und würde auf diese Weise ebenfalls multimodal vermittelt. Die Umsetzung der Anweisung kann dabei ebenfalls diskret oder kontinuierlich erfolgen, wie es bereits im entwickelten ATN vorgesehen ist. Auf dieser Ebene sind aber ebenso metaphorische Interpretationen denkbar. Die Sprache vermittelt dann den Kontext, eventuell eine gewünschte Farbänderung, die Geste überträgt den Wert der Änderungsgröße. Die Motion-Modifikatoren setzen daraufhin das metaphorische mapping um.

Letztendlich gibt es auf sensorischer Seite den Bedarf kabelungebundene Verfahren zu finden. Ein Ansatz zur Lösung dieser Aufgabe wurde bereits in einem Kooperationsprojekt begonnen und könnte in den folgenden Jahren die herkömmliche Sensorik ersetzen. Damit würde ein großes Akzeptanzproblem gelöst werden. Das Argument, den Benutzer von zusätzlichen Eingabegeräten zu befreien, stellt sich als äußerst schwach heraus, wenn man ihn dafür mit einem Baum von Spezialsensoren verkabelt. Schlussendlich hat die hier betrachtete Anwendung der VK beispielhaften Charakter, es sollten weitere Erprobung in anderen Domänen erfolgen (s.o.).

Diese Möglichkeiten bieten ein weites Spektrum von Folgebestrebungen. Virtuelle Umgebungen sind im Prinzip ja als ein Spezialfall einer Anwendung mit der Bereitstellung einer expliziten Raum- und Objektspezifikation in einem gemeinsamen Referenzsystem zu sehen. Löst man sich von diesem Gedanken und betrachtet man die kürzlich begonnene Entwicklung zum *wearable computing* und den *smart rooms*, so erweitert sich das Einsatzpotential erheblich. Die Fähigkeiten, Referenten und Aktionen zu bestimmen, sind in diesen Einsatzgebieten überall vorhanden. Externe Sensorik vorausgesetzt, könnte sich hier ein Einsatzfeld der beschriebenen Verfahren auf einem bisher als gänzlich unterschiedlich angesehenen Gebiet eröffnen.

Epilog

Diese Arbeit leistet mit den vorgelegten Ergebnissen einen Beitrag zur Erschaffung einer weitgehend natürlichen Mensch-Maschine-Kommunikation in und für VR-Anwendungen. Um diese Aufgabe zu erfüllen, wurden übertragbare benötigte Verfahren und Konzepte entwickelt, mit deren Hilfe man dem Gesamtziel näher kommt: Die Bedienung von immer komplexeren Systemen handhabbarer und menschengerechter zu gestalten. Leitend war hier die Motivation, einen Anwender nicht zur Erlernung einer vom System fest vorgegebenen Interaktionssprache unter Verwendung dedizierter Eingabegeräte zu zwingen, sondern Systeme mit den Möglichkeiten zu versehen, natürliche — sprachliche und gestische — Äußerungen im Situationskontext richtig zu interpretieren.

Literaturverzeichnis

[1] *Intelligent User Interfaces*, chapter Intelligent Multi-Media Interface Technology, pages 11–45. Addison-Wesley Publishing Company, 1991.

[2] James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, (23), 1984.

[3] M. Andre, V. G. Popescu, A. Shaikh, A. Medl, I. Marsic, C. Kulikowsky, and J. Flanagan. Integration of speech and gesture for multimodal human-computer interaction. pages 20–27, Tilburg, January 1998.

[4] C. Bailey-Kellog, F. Zhao, and K. Yip. Spatial Aggregation: language applications. In *Proceedings of AAAI*. AAAI, 1996.

[5] OpenGL Architecture Review Board. *The OpenGL Reference Manual*. Addison-Wesley Publishing Company, 1992.

[6] K. Böhm, W. Broll, and M. Sokolewicz. Dynamic gesture recognition using neural networks; a fundament for advanced interaction construction. In S. Fisher, J. Merrit, and M. Bolan, editors, *Stereoscopic Displays and Virtual Reality Systems, SPIE Conference Electronic Imaging Science & Technology*, volume 2177, San Jose, USA, 1994.

[7] K. Böhm, W. Hübner, and K. Väänänen. Given: Gesture driven interactions in virtual environments; a toolkit approach to 3D interactions. In *Interfaces to Real and Virtual Worlds*, 1992.

[8] R. A. Bolt. Put-That-There: Voice and gesture at the graphics interface. In *ACM SIGGRAPH—Computer Graphics*, New York, 1980. ACM Press.

[9] R. A. Bolt and E. Herranz. The integrated multi-modal interface. In *Proceedings of UIST '92, Fifth Annual Symposium on User Interface Software and Technology*, pages 15–18, Monterey, CA, November 1992.

[10] Sven Bormann. *Virtuelle Realität: Genese und Evaluation.* Addison-Wesley, Reading, MA, USA, 1. edition, 1994.

[11] Annelies Braffort, Rachidet Gherbi, Sylvie Gibet, and Daniel Richardson, editors. *Gesture-based Communication in Human-Computer Interaction - Proceedings of the International Gesture Workshop 1999, Gif-sur-Yvette*, number 1739 in Lecture Notes in Artificial Intelligence. Springer-Verlag, March 1999.

[12] B. Butterworth and G. Beattie. Gesture and silence as indicators of planning in speech. In Campbell and Smith, editors, *Recent Advances in the Psychology of Language.* Plenum Press, New York, 1978.

[13] Antonio Camurri and Pasqualino Ferrentino. Interactive environments for music and multimedia. *Multimedia Systems*, 7:32–47, 1999.

[14] Yong Cao, Bernhard Jung, and Ipke Wachsmuth. Situated verbal interaction in virtual design and assembly. In *Proc. Fourteenth International Joint Conference on Artificial Intelligence*, 1995.

[15] Rikk Carey, Gavin Bell, and Chris Marrin. Iso/iec 14772-1:1997 virtual reality modeling language (VRML). Technical report, The VRML Consortium Incorporated, 1997.

[16] Justine Cassell. *Embodied conversational agents: a new paradigm for the study of gesture and for human-computer interface*, chapter 11, pages 203–222. Oxford University Press, 1999.

[17] Marc Cavazza, Xavier Pouteau, and Didier Pernel. Multimodal communication in virtual environments. In *Symbiosis of Human and Artifact*, pages 597–604. Elsevier Science B. V., 1995.

[18] PHIGS+ Committee and Andries van Dam (chair). PHIGS+ functional description, revision 3.0. *Computer Graphics*, 22(3):125–218, July 1988.

[19] C. Cruz-Neira, D.J. Sandin, and T.A. DeFanti. Surround-screen projection based virtual reality: The design and implementation of the cave. In *Computer Graphics Proceedings, Annual Conference Series 1993*, pages 135–142. ACM SIGGRAPH, 1993.

[20] F. Cummins and R.F. Port. Rhythmic commonalities between hand gestures and speech. In *Proceedings of the 18. Meeting of the Cognitive*

Science Society, pages 415–419, Hillsday N.J., 1996. Lawrence Erlbaum Associates.

[21] Jan-Peter de Ruiter. *Gesture and Speech Production.* Number 6 in MPI Series in Psycholinguistics. Ponsen & Looijen bv, Wageningen, 1998.

[22] D. Efron. *Gesture and Environments.* King's Crown Press, Morningside Hights, New York, 1941.

[23] Paul Ekman and Wallace V. Friesen. Handbewegungen. In Klaus R. Scherer and Harald G. Wallbott, editors, *Nonverbale Kommunikation: Forschungsberichte zum Interaktionsverhalten.* Beltz Verlag, Weinheim und Basel, 1979.

[24] G. A. Fink, C. Schillo, F. Kummert, and G. Sagerer. Incremental speech recognition for multimodal interfaces. [35], pages 2012–2017.

[25] James Foley, Andries van Dam, Steven Feiner, and John Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley Systems Programming Series. Addison-Wesley, Reading, MA, USA, 1990.

[26] Martin Fröhlich. *Ein wissensbasiertes Rahmensystem zur merkmalsbasierten Gestenerkennung für multimediale Anwendungen.* PhD thesis, Technische Fakultät, Universität Bielefeld, 1999.

[27] K.S. Fu, R.C. Gonzalez, and C.S.G. Lee. *Robotics: control, sensing, vision and intelligence.* McGraw-Hill, Singapore, 1987.

[28] Salvatore Gaglio, Aldo Grattarola, Lina Massone, and Pietro Morasso. *Human Movement Understanding*, chapter Analysis and synthesis of smooth shapes, pages 247–284. Volume 33 of Morasso and Tagliasco [70], 1986.

[29] E. Giachin and S. McGlashan. Spoken language dialog systems. In K. Church, S. Young, and G. Bloothooft, editors, *Corpus-based Methods in Language and Speech Processing.* Kluwer, 1996.

[30] O. J. Grüsser and U. Grüsser-Cornehls. Physiologie des Sehens. In *Grundriss der Sinnesphysiologie*, pages 151–216. Springer-Verlag, Berlin, 1980.

[31] Philip A. Harling and Alistair D. N. Edwards. Hand tension as a gesture segmentation cue. pages 75–87, Berlin Heidelberg New York, 1997. Dep. of Computer Science, University of York, Springer-Verlag.

[32] A.G. Hauptmann and P. McAvinney. Gestures with speech for graphic manipulation. *International Journal of Man-Machine Studies*, 38:231–249, 1993.

[33] Martin Hoffenke and Ipke Wachsmuth. Object recognition with shape prototypes in a 3D construction scenario. In W. Burgard, A.B. Cremers, and T. Christaller, editors, *Advances in Artificial Intelligence, Proceedings KI-99*, Lecture Notes in Artificial Intelligence, pages 231–242. Springer-Verlag, 1999.

[34] Martin Hoffhenke, Bernhard Jung, and Stefan Kopp. Der Cody Virtuelle Konstrukteur: Manual version 2.0. Technical report, SFB 360, Universität Bielefeld, 1998.

[35] IEEE. *IECON'98: Proceedings of the 24th annual Conference of the IEEE Industrial Electronics Society*, Aachen, September 1998. IEEE Computer Society Press.

[36] M. Johnston. Unification-based multimodal parsing. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 98)*, 1998.

[37] Michael Johnston, Philip R. Cohen, David McGee, Sharon L. Oviatt, James A. Pittman, and Ira Smith. Unification-based multimodal integration. In *35th Annual Meeting of the Association for Computational Linguistics, Madrid*, pages 281–288, 1997.

[38] Bernhard Jung. *Wissensverarbeitung für Montageaufgaben in virtuellen und realen Umgebungen.* PhD thesis, Technische Fakultät der Universität Bielefeld, 1996.

[39] Bernhard Jung, Martin Hoffenke, and Ipke Wachsmuth. Virtual assembly with construction kits. In *Proceedings of the 1998 ASME Design for Engineering Technical Conferences (DECT-DFM '98)*, 1998.

[40] Bernhard Jung, Stefan Kopp, Marc Latoschik, Timo Sowa, and Ipke Wachsmuth. Virtuelles Konstruieren mit Gestik und Sprache. *Künstliche Intelligenz*, 2:5–11, 2000.

[41] Bernhard Jung, Marc Erich Latoschik, and Ipke Wachsmuth. Knowledge-based assembly simulation for virtual prototype modeling. [35], pages 2152–2157.

[42] Adam Kendon. Gesticulation and speech: Two aspects of the process of utterance. In M. R. Key, editor, *The Relation between Verbal and Non-verbal Communication*. Mouton, The Hague, 1980.

[43] Adam Kendon. Current issues in the study of gestures. In J.-L. Nespoulous, P. Rerron, and A.R. Lecours, editors, *The Biological Foundations of Gestures: Motor and Semiotic Aspects*. Lawrence Erlbaum Associates, Hillsday N.J., 1986.

[44] Adam Kendon, Thomas A. Sebeok, and Jean Umiker-Sebeok, editors. *Nonverbal Communication, Interaction, and Gesture: Selections from Semiotica*, volume 41 of *Approaches to Semiotics*. Mouton, The Hague, 1981.

[45] D.B. Koons, C.J. Sparrel, and K.R. Thorisson. Intergrating simultaneous input from speech, gaze and hand gestures. In *Intelligent Multimedia Interfaces*. AAAI Press, 1993.

[46] Stefan Kopp. Ein wissensbasierter Ansatz zur Modellierung von Verbindungen zur Virtuellen Montage. Master's thesis, Technische Fakultät, Universität Bielefeld, 1998.

[47] Myron W. Krueger. *Artificial Reality II*. Addison-Wesley, Reading, MA, USA, 1990.

[48] W. Krüger, C.A. Bohn, B. Fröhlich, H. Schüth, W. Strauss, and G. Wesche. The responsive workbench: A virtual work environment. *IEEE Computer*, 28(7), 1995.

[49] Marc Erich Latoschik. A general framework for multimodal interaction in virtual reality systems: PrOSA. In *VR2001 workshop proceedings: The Future of VR and AR Interfaces: Multi-modal, Humanoid, Adaptive and Intelligent*, 2001. in press.

[50] Marc Erich Latoschik, Martin Fröhlich, Bernhard Jung, and Ipke Wachsmuth. Utilize speech and gestures to realize natural interaction in a virtual environment. [35], pages 2028–2033.

[51] Marc Erich Latoschik, Bernhard Jung, and Ipke Wachsmuth. Multimodale Interaktion mit einem System zur Virtuellen Konstruktion. In Klaus Beiersdörfer, Gregor Engels, and Wilhelm Schäfer, editors, *Proc, der 29. Jahrestagung der Gesellschaft für Informatik - Informatik'99, Informatik überwindet Grenzen*, pages 88–97, Berlin Heidelberg New York, 1999. Springer-Verlag.

[52] Marc Erich Latoschik and Ipke Wachsmuth. Exploiting distant pointing gestures for object selection in a virtual environment. In Wachsmuth and Fröhlich [95], pages 185–196.

[53] Marc Erich Latoschik and Ipke Wachsmuth. Sprachgestützte gestische Interaktion zur Steuerung Virtueller Konstruktion. In *Tagungsband zum Workshop des Forschungsverbundes NRW – Die Virtuelle Wissensfabrik – vom 23./24. September 1999*, GMD, Schloß Birlinghoven, St. Augustin, 1999.

[54] Britta Lenzmann. Spezifikation und Implementierung eines Mehragentensystems zur Interaktion mit einer virtuellen Umgebung. Master's thesis, Technische Fakultät der Universität Bielefeld, AG Wissensbasierte Systeme, Universitätsstraße 25, 33615 Bielefeld, March 1994.

[55] Britta Lenzmann. *Benutzeradaptive und multimodale Interface-Agenten.* PhD thesis, Technische Fakultät, Universität Bielefeld, 1998.

[56] Willem J.M. Levelt, Graham Richardson, and Wido La Heu. Pointing and voicing in deictic expressions. *Journal of Memory and Language*, 1985.

[57] Mark Lucente, Gert-Jan Zwart, and Andrew D. George. Visualization space: A testbed for deviceless multimodal user interface. In *Intelligent Environments Symposium*, American Assoc. for Artificial Intelligence Spring Symposium Series, March 1998.

[58] I. Scott MacKenzie. Input devices and interaction techniques for advanced computing. In *Virtual Environments and Advanced Interface Design.* Oxford University Press, 1995.

[59] Pattie Maes. Alive: An artificial live interactive video environment. In Thomas E. Linehan, editor, *Computer Graphics Visual Proceedings*, Annual Conference Series, page 189, New York, NY 10036, USA, 1993. ACM Press, ACM SIGGRAPH.

[60] Pattie Maes. Atrificial life meets entertainment: Lifelike autonomous agents. *Special Issue on Novel Applications of AI Communications of the Association for Computing Machinery*, 1995.

[61] Pattie Maes, Trevor Darell, Bruce Blumberg, and Alex Pentland. The alive system: Full-body interaction with autonomous agents. In *Proceedings of Computer Animation Conference (April 19-21)*, Geneva, Switzerland, 1995. IEEE.

[62] C. Manetta and R. Blade. Glossary of virtual reality terminology. *International Journal of Virtual Reality*, 1(2), 1995.

[63] Mark T. Maybury. Research in multimedia an multimodal parsing and generation. In P. McKevitt, editor, *Journal of Artificial Intelligence Review: Special Issue on the Integration of Natural Language and Vision Processing*, volume 9, pages 2–27. Kluwer Academic Publishers Group, 1993.

[64] Scott McGlashan. Speech interfaces to virtual reality. In *Proceedings of the Second Conference on the Military Application of Synthetic Environments and Virtual Reality*, Stockholm, Schweden, 1995.

[65] D. McNeill. *Hand and Mind: What Gestures Reveal about Thought.* University of Chicago Press, Chicago, 1992.

[66] D. McNeill and E. Levy. Conceptual representations in language activity and gesture. In Jarvella and Klein, editors, *Speech, Place, and Action*, pages 271–295. John Wiley, New York, NY, USA, 1982.

[67] David McNeill. Triangulating the growth point – arriving at consciousness. In Lynn S. Messing and Ruth Campbell, editors, *Gesture, Speech, and Sign.* Oxford University Press, 1999.

[68] Mark R. Mine. Isaac: a meta-cad system for virtual environments. *Computer-aided Design*, 29(8):547–553, 1997.

[69] Pietro Morasso. *Human Movement Understanding*, chapter Trajectory Formation, pages 9–58. Volume 33 of Morasso and Tagliasco [70], 1986.

[70] Pietro Morasso and Vincenzo Tagliasco, editors. *Human Movement Understanding*, volume 33 of *Advances in Psychology.* Elsevier Science Publishers B. V., (North-Holland), 1986.

[71] Palmer Morrel-Samuals and Robert M. Krauss. Word familarity predicts temporal asynchrony of hand gesture and speech. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 18(3):615–622, 1992.

[72] Axel Mulder. Human movement tracking technology. Technical Report 94-1, School of Kinesiology, Simon Fraser University, July 1994.

[73] Jackie Neider, Tom Davis, and Mason Woo. *The OpenGL Programming Guide.* Addison-Wesley Publishing Company, 1993.

[74] J.-L. Nespoulous and A.R. Lecours. Gestures: Nature and function. In J.-L. Nespoulous, P. Rerron, and A.R. Lecours, editors, *The Biological Foundations of Gestures: Motor and Semiotic Aspects*. Lawrence Erlbaum Associates, Hillsday N.J., 1986.

[75] Claudia Nölker and Helge Ritter. Detektion von Fingerspitzen in Videobildern. In *DAGM Mustererkennung 1997*, pages 97–104. Springer-Verlag, 1997.

[76] Claudia Nölker and Helge Ritter. Detection of fingertips in human hand movement sequences. In Wachsmuth and Fröhlich [95], pages 209–218.

[77] Claudia. Nölker and Helge Ritter. Illumination independent recognition of deictic arm postures. [35], pages 2006–2011.

[78] Claudia Nölker and Helge Ritter. Grefit: Visual recognition of hand postures. In Braffort et al. [11].

[79] International Standards Organisation. International standard information processing systems – computer graphics – Graphical Kernel System for three dimensions (GKS-3D). Technical Report 8805, ISO Document, 1988.

[80] http://www.opensg.org. WWW.

[81] B. Rimé and L. Schiaratura. Gesture and speech. In Feldman and Rimé, editors, *Fundamentals of Nonverbal Behavior*. Press Syndicate of the University of Cambridge, New York, 1991.

[82] Ernst Günter Schukat-Talamazzini. *Automatische Spracherkennung, Statistische Verfahren der Musteranalyse*. Vieweg Verlag, 1995.

[83] SGI. *IRIS Performer Programmer's Guide*.

[84] B. Sherman and P. Judkins. *Glimpses of Heaven, Visions of Hell: Virtual Reality and its implications*. Hodder and Stoughton, London, 1992.

[85] S. Sherr. *Input Devices*. Academic Press, 1988.

[86] Timo Sowa, Martin Fröhlich, and Marc Erich Latoschik. Temporal symbolic integration applied to a multimodal system using gestures and speech. In Annelies Braffort, Rachid Gherbi, Sylvie Gibet, James

Richardson, and Daniel Teil, editors, *Gesture-Based Communication in Human-Computer Interaction - Proceedings of the International Gesture Workshop (Gif-sur-Yvette, France, March 1999)*, Lecture Notes in Artificial Intelligence, pages 291–302, Berlin Heidelberg New York, 1999. Springer-Verlag.

[87] Carlton J. Sparrell and David B. Koons. Interpretation of coverbal depictive gestures. In *AAAI Spring Symposium Series*, pages 8–12. Stanford University, March 1994.

[88] R.K. Srihari. Computational models for integrating linguistic and visual information: A survey. *Artificial Intelligence Review*, 8:349–369, 1995.

[89] Paul S. Strauss and Rikk Carey. An object-oriented 3D graphics toolkit. In *Computer Graphics*, volume 26 of *SIGGRAPH Proceedings*, pages 341–349, 1992.

[90] K. R. Thorisson. *Communicative Humanoids*. PhD thesis, School of Architecture and Planning, Massachussetts Institute of Technology, September 1996.

[91] Henrik Tramberend. A distributed virtual reality framework. In *Virtual Reality*, 1999.

[92] Didier Verna and Alain Grumbach. Can we define Virtual Reality? The MRIC Model. In *Virtual Worlds 98*, number 1434 in Lecture Notes in Artificial Intelligence, LNAI, pages 29–41, 1998.

[93] M.T. Vo and C. Wood. Building an application framework for speech and pen input integration in multimodal learning interfaces. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA, 1996.

[94] Ian Voss. Empirische Untersuchungen zu sprachbegleitender Gestik im Mensch-Maschine-Dialog. Master's thesis, Technische Fakultät, Universität Bielefeld, 1999.

[95] Ipke Wachsmuth and Martin Fröhlich, editors. *Gesture and Sign-Language in Human-Computer Interaction: Proceedings of Bielefeld Gesture Workshop 1997*, number 1371 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg New York, 1998. Springer-Verlag.

[96] Ipke Wachsmuth, Ian Voss, Marc Erich Latoschik, Stefan Kopp, and Bernhard Jung. Multimodale Interaktion in der Virtuellen Realität. In Horst Oberquelle, Reinhard Oppermann, and Jürgen Krause, editors, *Mensch & Computer 2001*, number 55 in Berichte des German chapter of the ACM, pages 265–274. ACM, GI, B.G. Teubner Stuttgart, 2001.

[97] http://www.web3d.org. WWW.

[98] D. Weimer and S.K. Ganapathy. Interaction techniques using hand tracking and speech recognition. In M.M. Blattner and R.B. Dannenberg, editors, *Multimedia Interface Design*, pages 109–126. ACM Press, 1992.

[99] Josie Wernecke. *The Inventor Mentor, Programming Object-Oriented 3D Graphics with Open Inventor, Release 2.* Addison-Wesley Publishing Company, 1994.

[100] A. D. Wexelblat. A feature-based approach to continous-gesture analysis. Master's thesis, Massachusetts Institute of Technology, Advanced Human Interface Group, Cambridge, MA, USA, 1994.

[101] Alan Daniel Wexelblat. An approach to natural gesture in virtual environments. *acm Transactions on Computer-Human Interaction*, 2(3):179–200, 1995.