



TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

Lehrstuhl für Echtzeitsysteme und Robotik

Intelligente Steuerung von automatischen
Kamerasystemen in Nachrichtensstudios durch
verbale Kontrolle

Sebastian T. Frühling

Vollständiger Abdruck der von der Fakultät der Informatik der Technischen Universität München
zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. rer. nat. Johann Schlichter

Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing., Dr.-Ing. habil. Alois Knoll

2. Univ.-Prof. Dr. rer. nat. Bernd Radig

Die Dissertation wurde am 23.10.2014 bei der Technischen Universität München eingereicht
und durch die Fakultät für Informatik am 02.03.2015 angenommen.

Abstract

With the growing impact of cost-pressure and the perspective of new, unseen video effects in broadcasting environments the interest on automatic camera systems increase. Current systems are able to engage previously stored positions and run predefined moves. An operator adjusts these positions to the current circumstances in the studio environment. In this thesis, a novel intelligent control-system for automated cameras is presented, that integrates best-possible into an existing studio environment and that builds the base for an autonomous usage in broadcasting environments. The system extends the Camerobot RoboKam[©] and can, due to the usage of axis-speed-control, easily be adapted to other systems. The requirements of such a system has been elaborated with the help of theoretical and practical analysis. As a key feature of such a system the voice command functionality has been identified. This kind of voice control as well as automatic adjustment of shots for unforeseen changes in the environment has never been supported by any systems before. The presented control unit interprets commands in natural language by using a combinatory categorial grammar. The commands are converted into a logical expression and then into an appropriate movement of the camera. The particular actions correlate to the actions a human camera-operator would execute. These actions contain joystick-actions to move the camera as well as actions to adapt a previously saved shot to current circumstances. The excellence of the system has been rated in a survey among studio staff.

Zusammenfassung

Mit zunehmenden Kostendruck und der Aussicht auf neuartige Video-Effekte wächst in der Fernsehwelt das Interesse an automatischen Kamerasystemen. Aktuelle Systeme sind in der Lage zuvor gespeicherte Positionen anzufahren und gespeicherte Fahrten durchzuführen. Ein Operator passt die gespeicherten Positionen und Fahrten auf die aktuellen Gegebenheiten im Studio an. In dieser Arbeit wird eine neuartige, intelligente Steuerung für Kamerasysteme vorgestellt, die sich bestmöglich in die bestehende Studio-Welt einfindet und die Grundlage für den autonomen Einsatz im Nachrichtenstudio bildet. Das System wurde auf der Basis der Came-robot RoboKam[©] entwickelt und kann, durch die Verwendung einer allgemeinen Schnittstelle auf der Basis von Achsengeschwindigkeiten, leicht auf andere Systeme übertragen werden. Anhand von theoretischen und praktischen Nachweisen wurden die Anforderungen für ein solches System ausgearbeitet. Das Verstehen von Kommandos in natürlicher Sprache wurde als Schlüsselfunktion eines solchen Systems identifiziert. Diese Form der Kommunikation unterstützen bisherige Kamerasysteme ebenso wenig wie die automatische Kompensation von Fehlkadrierungen. Das vorgestellte Kontrollsystem interpretiert Kommandos in natürlicher Sprache anhand einer entwickelten kombinatorischen Kategorial-Grammatik. Die Kommandos werden durch die Grammatik in einen logischen Ausdruck übersetzt und daraufhin in eine entsprechende Aktion des Kamerasystems umgewandelt. Die jeweiligen Aktionen entsprechen denen, die ein menschlicher Kameramann ebenfalls durchführen würde. Darunter fallen Joystick-Aktionen und Ausgleichsaktionen, um ein geforder-tes Kamerabild der aktuellen Situation im Studio anzupassen. Die Güte des Systems wurde im Rahmen einer Feldstudie unter Berufsträgern verifiziert.

Danksagung

Diese Arbeit entstand im Rahmen meiner Tätigkeit als Software-Entwickler bei der Firma Robotics Technology Leaders GmbH (RTLeaders), deshalb gilt mein Dank den Gesellschaftern von RTLeaders Prof. Knoll sowie Dr. Riesner. Ohne Sie wäre die Arbeit so nicht möglich gewesen. Des Weiteren möchte ich mich bei Prof. Radig für die hilfreichen Tipps und das Erstellen des Zweitgutachtens bedanken.

Die Kollegen von RTLeaders müssen an dieser Stelle ebenfalls erwähnt werden. Viele Probleme konnte ich in diesem Rahmen diskutieren und die Arbeit wäre ohne die gute Zusammenarbeit mit Jun Yuan und Stephan Grune wohl nie fertig geworden.

Mein Dank geht auch an alle Interviewpartner. Sie haben sich oft während einer Sendung oder in ihrer Freizeit Zeit für mich genommen und sich bemüht mir meine Fragen bestmöglich zu beantworten.

Den Kollegen der Avirtico GmbH, im besonderen Michael Muschler, gehört ebenfalls mein Dank. Der beständige Quell von Fachwissen, viele Diskussionen und wichtige Hilfe beim Einrichten des Render-Systems haben mich oft motiviert weiter zu machen.

Dr. Manuel Giuliani von der Fortiss GmbH gehört ebenfalls mein Dank. Er war mir ein guter und hilfsbereiter Ansprechpartner in Sachen Grammatiken. Außerdem möchte ich mich bei Alexander Bannat bedanken. Seine Korrekturen und Tipps waren ebenfalls ausgesprochen hilfreich.

Zu guter Letzt danke ich meiner Familie, die mich während dieser Zeit ertragen musste. Insbesondere meinem Vater, der mir stets ein lieb gewonnener Diskussionspartner war sowie natürlich meiner Frau Simone, die mir stets verständnisvoll zur Seite stand.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
1 Einführung	1
1.1 Produktion von Nachrichtensendungen	2
1.2 Problematik automatischer Kamerasysteme	3
1.3 Ziele und Aufbau der Arbeit	4
2 Stand der Technik und wissenschaftlicher Überblick	7
2.1 Kamerasysteme beim Film und Fernsehen	7
2.1.1 Manuelle Kameraführung	9
2.1.2 Programmierte Kameraführung	9
2.2 Arbeiten mit Bezug auf Kamera-Einstellungen	10
2.3 Arbeiten mit Bezug auf Kamerafahrten	13
2.4 Kamerasysteme in medizinischen Umgebungen	16
2.5 Sprachgeführte Systeme	18
2.6 Kombinatorische Kategorial-Grammatik (CCG)	23
2.6.1 Historie	24
2.6.2 Aufbau und Funktionsweise der Kategorial-Grammatik	24
2.6.2.1 Slash-Modalitäten	25
2.6.2.2 Ableitungsregeln	26
2.6.3 Semantik in der CCG	28
2.6.4 OpenCCG	30
2.6.4.1 Eigenschaften von Wörtern in OpenCCG - features	30
2.6.4.2 Zuweisung von Kategorien zu Wörtern - words	31
2.6.4.3 Hybrid-Logik zum Ausdruck semantischer Information	32
2.6.4.4 Vom Satz zum logischen Ausdruck	33

INHALTSVERZEICHNIS

3	Ausgewählte Aspekte bei Nachrichten- und Magazinsendungen	37
3.1	Theoretische Aspekte der Kameraführung	38
3.1.1	Kameraführung im Wandel der Zeit	38
3.1.2	Einstellungsgrößen	39
3.1.3	Kamerablickwinkel	42
3.1.4	Safe-Title und Safe-Action	43
3.1.5	Kamerabewegungen und Kamerafahrten	44
3.2	Praktische Aspekte der Kameraführung aus Sicht der Regie	47
3.2.1	Kommunikation zwischen Regie und Kamera	47
3.2.2	Ablauf einer Sendung am Beispiel von <i>Zur Sache Baden-Württemberg</i>	49
3.2.3	Ergebnisse der Feldstudie zur praktischen Kameraarbeit	51
3.2.3.1	Typische Anweisungen aus der Regie	52
3.2.3.2	Merkmale eines guten Bildes	53
3.2.3.3	Merkmale eines guten Kameramanns	54
3.2.3.4	Genre-spezifische Merkmale	55
3.2.3.5	Merkmale einer guten Kamerafahrt	56
3.3	Vergleich der Nachrichten von ARD, ZDF und RTL aus der Sicht des Zuschauers	57
3.3.1	Einstellungsfolge während einer Sendung	57
3.3.2	Position und Größe des Hauptmoderators während einer Sendung	61
3.3.3	Zusammenfassung der Auffälligkeiten der Einstellungsfolgen und der Einstellungsgrößen	64
4	Anforderungen an ein intelligentes automatisches Kamerasystem	67
4.1	Multiple Aktionen	67
4.2	Natürlichsprachliche Kommandos	69
4.3	Hohe Flexibilität	71
4.3.1	Variabilität der Arbeitsbegriffe	71
4.3.2	Kompensationsstrategie	72
5	Automatische Verarbeitung von natürlicher Sprache im Studio	73
5.1	Stopp-Kommando	76
5.2	Joystick-Kommandos	77
5.2.1	Bewegungsachsen	78

5.2.2	Richtung durch richtungsbeschreibende Adverben	80
5.2.3	Richtung durch Komparation von Adjektiven	82
5.2.4	Richtung durch ortsbeschreibende Adverben	84
5.2.5	Unbestimmte Quantität	85
5.2.6	Bestimmte Quantität	87
5.3	Objektbezogene Kommandos	88
5.3.1	Bildausschnittsbestimmende Kommandos	88
5.3.2	Kommandos zur Positionierung von Objekten im Bild	91
5.3.3	Kommandos zur Raumänderung von Objekten	93
5.4	Konditional-Kommandos	95
5.5	Speichern-Kommando	98
5.6	Anfahrt einer bestimmten Einstellung	101
5.7	Zeitwährende Kommandos	103
6	Das Aktionsmodell zur Handlungsgenerierung	105
6.1	Der Regelkreis	107
6.1.1	Percept	108
6.1.2	Reflect	108
6.1.3	Act	108
6.2	Reflect – eine hierarchische Regelstruktur	108
6.2.1	General	110
6.2.2	Interpretation	111
6.2.3	Specialization	111
6.2.4	Goal-Comparison	111
6.2.5	Goal-Adaption	112
6.2.6	Joystick-Execution	112
6.3	Stopp-Kommando	113
6.4	Joystick-Kommandos	114
6.4.1	Fahrdistanz	116
6.4.2	Zurück / Weiter	116
6.5	Bildausschnittsbestimmende Kommandos	118
6.6	Kommandos zur Positionierung von Objekten	121
6.7	Kommandos zur Raumänderung von Objekten	122
6.8	Konditional-Kommandos	124

INHALTSVERZEICHNIS

6.9	Speichern-Kommando	126
6.10	Anfahren einer bestimmten Positionen und Einstellung	126
6.11	Zeitwährende Kommandos	129
7	Implementierung und Evaluation des Prototyps HULICO	131
7.1	Prototyp: HULICO	131
7.1.1	RoboKam [©]	133
7.1.2	Schematischer Aufbau	134
7.1.3	Realisierung	135
7.2	Eigenschaften des Systems	137
7.2.1	Joystick-Aktionen	138
7.2.2	Laden und Speichern	140
7.2.3	Zeitwährende Kommandos	141
7.2.4	Bildfokussierende Fahrten	142
7.2.5	Flexibilität des Prototyps	143
7.3	Evaluation des Prototyps im Rahmen einer Benutzerstudie	145
7.3.1	Ablauf der Befragung	146
7.3.2	Teilnehmer der Befragung	147
7.3.3	Verwendeter Fragebogen	147
7.3.4	Hypothesen der Benutzerstudie	149
7.3.5	Ergebnisse der Befragung	150
7.3.6	Diskussion	155
7.3.6.1	Diskussion der Ergebnisse - Teil 1	155
7.3.6.2	Diskussion der Ergebnisse - Teil 2	159
8	Zusammenfassung und Ausblick	163
8.1	Das Ziel dieser Arbeit	163
8.2	Die Inhalte dieser Arbeit	163
8.3	Das Ergebnis dieser Arbeit	164
A	Appendix	167
	Literaturverzeichnis	173

Abbildungsverzeichnis

1.1	Schematischer Ablauf einer Nachrichtensendung	3
1.2	Gegenüberstellung einer fehlerhaften mit einer richtig kadrierten Einstellungen	4
2.1	Darstellung der RoboKam7 und eines studiotypischen Bedienpaneels	8
2.2	Abbildung des Multiplayer-Computerspiels von Tomlinson et al.	13
2.3	Formale Beschreibung des Vertigo-Effekts in CINEMA	14
2.4	ShowMotion von Burtnyk et al.	15
2.5	Medizinische kameratragende Systeme	16
2.6	Voice-Control-System von Holada und Pelc.	20
2.7	Sprachgeführte System von Tellex et al.	22
2.8	Slash-Modalitäten von Ableitungsregeln	25
2.9	Ableitung von <i>Die Katze mag die Couch</i> mit Hilfe der CCG	28
2.10	Ableitung von <i>mag die Couch</i> anhand der CCG	29
2.11	Typische Features für deutsche Grammatiken	31
2.12	Die Definition des Wortes Katze in OpenCCG	32
2.13	Einfaches Beispiel zur Verwendung der DotCCG	33
2.14	Durch OpenCCG generierte logische Form	34
2.15	Durch OpenCCG generierte XML-Struktur	35
3.1	Schematische Darstellung der Haupteinstellungsgrößen im Nachrichtenbereich.	40
3.2	Unterschiedliche Wirkungen von Einstellungen	43
3.3	Safe-Title und Safe-Action im VideoBild	44
3.4	Vergleich von <i>Totalen-Einstellungen</i> durch Zoom und Fahrt.	45
3.5	Fixpunkt bei einer Zoomfahrt	46
3.6	Eröffnungssequenz von <i>Zur Sache Baden-Württemberg</i> am 22.09.2011	50

ABBILDUNGSVERZEICHNIS

3.7	Einstellungs-Verlauf der <i>tagesschau</i> -Sendung am 18.02.2011	58
3.8	Einstellungs-Verlauf der <i>heute</i> -Sendung am 18.02.2011 - Teil 1	59
3.9	Einstellungs-Verlauf der <i>heute</i> -Sendung am 18.02.2011 - Teil 2	60
3.10	Einstellungs-Verlauf der <i>RTL Aktuell</i> -Sendung am 18.02.2011	60
3.11	Vergleich der <i>Nahe-Einstellungen</i> in den Nachrichten von RTL und ZDF	62
3.12	Vergleich der <i>Halbnahe-Einstellungen</i> in den Nachrichtensendungen von ARD, RTL und ZDF	63
3.13	Vergleich der <i>Amerikanischen-</i> und <i>Zweier-Einstellungen</i> von <i>RTL Aktuell</i> und <i>ZDF heute</i>	64
5.1	Basis Struktur eines Studio-Kommandos	74
5.2	Struktur eines Stopp-Kommandos.	76
5.3	Ableitung des Stopp-Kommandos durch CCG.	76
5.4	Logische Form des Stopp-Kommandos.	77
5.5	Beispiele für positionsändernde Joystick-Kommandos.	78
5.6	Beispiele für orientierungsändernde Joystick-Kommandos.	79
5.7	Beispiele für linsensteuernde Joystick-Kommandos.	79
5.8	Struktur der Joystick-Kommandos	80
5.9	Ableitung eines richtungsbeschreibenden Adverbs.	81
5.10	Logische Form von Joystick-Kommandos, deren Richtung durch ein richtungs- beschreibendes Adverb bestimmt wird.	81
5.11	Ableitung eines Adjektivs im Komparativ.	82
5.12	Logische Form von <i>Kamera1 fahre tiefer</i>	82
5.13	Ableitung eines Komparativs unter Verwendung von <i>weiter</i>	83
5.14	Logische Form von <i>Kamera1 fahre weiter weg</i>	83
5.15	Ableitung eines Joystick-Kommandos mit Hilfe der CCG.	84
5.16	Logische Form eines Joystick-Kommandos.	85
5.17	Zusammengesetzte Quantitätsangabe	86
5.18	Beispiele für Fahr-Kommandos mit bestimmter Quantität.	87
5.19	Struktur der Joystick-Kommandos mit bestimmter Quantität.	87
5.20	Ableitung eines Satzes mit bestimmter Quantität.	87
5.21	Logische Form eines Joystick-Kommandos in Verbindung mit bestimmter Quantität.	88

5.22 Beispiele für ausschnittsbestimmende Kommandos.	89
5.23 Struktur der bildausschnittsbezogenen Kommandos.	89
5.24 Ableitung eines bildausschnittsbestimmenden Kommandos.	90
5.25 Logische Form eines bildausschnittsbestimmenden Kommandos.	90
5.26 Beispiele zum Verschieben von Objekten im Bild	91
5.27 Struktur des Compose-Kommandos	92
5.28 Ableitung eines Compose-Kommandos.	92
5.29 Logische Form eines Compose-Kommandos.	93
5.30 Beispiele zum Ändern von Objekt-Eigenschaften.	93
5.31 Struktur der Kommandos zum Ändern von Objekt-Eigenschaften.	93
5.32 Ableitung eines modifyObject-Kommandos.	94
5.33 Logische Form eines modifyObject-Kommandos.	95
5.34 Beispiele für Konditional-Kommandos.	95
5.35 Struktur der Konditional-Kommandos.	96
5.36 Ableitung eines Konditional-Kommandos.	96
5.37 Logische Form eines Konditional-Kommandos.	97
5.38 Beispiele für das Speichern der aktuellen Position.	98
5.39 Struktur des Speichern-Kommandos.	98
5.40 Kategorien für das Verb <i>speichern</i>	99
5.41 Ableitung eines Speichern-Kommandos mit Hilfe der CCG.	100
5.42 Logische Form eines Speichern-Kommandos.	100
5.43 Beispiele zur Anfahrt einer Einstellung.	101
5.44 Struktur von Kommandos zur Anfahrt einer Einstellung.	101
5.45 Ableitung eines GotoShot-Kommandos.	102
5.46 Logische Form eines GotoShot-Kommandos.	103
5.47 Beispiele für zeitwährende Kommandos.	104
5.48 Logische Form eines zeitwährenden Kommandos.	104
6.1 Modell zur intelligenten Kamerasteuerung	106
6.2 Regelkreis zur Kameraführung	107
6.3 Hierarchie-Stufen im <i>Reflect</i> -Zustand	109
6.4 Repräsentation der 2D-Prädikate	110
6.5 Schematischer Ablauf eines Stopp-Kommandos	114

ABBILDUNGSVERZEICHNIS

6.6	Schematischer Ablauf einer Joystick-Anweisung	115
6.7	Schematischer Ablauf eines verringerten Zurück-Kommandos	117
6.8	Schematischer Ablauf eines bildausschnittsbestimmenden Kommandos	119
6.9	Schematischer Ablauf eines raumändernden Kommandos	122
6.10	Schematischer Ablauf einer Konditional-Anweisung	125
6.11	Schematischer Ablauf für eine Fahrt auf eine Einstellung	128
6.12	Schematischer Ablauf eines zeitwährenden Kommandos	130
7.1	Grafische Oberfläche des Prototyps HULICO.	132
7.2	Komponentendiagramm von HULICO	135
7.3	Klassenhierarchie der Aktions-Klassen	136
7.4	Unterschiedliche Endpositionen für studiotypische Joystick-Aktionen	138
7.5	Unterschiede für unterschiedlich quantifizierte Schwenks nach links	139
7.6	<i>Zurück</i> als Eigenschaft des Joysticks	139
7.7	Vergleich von Einstellungen für Moderatoren unterschiedlicher Körpergröße	140
7.8	Beispiel einer <i>Totale-Einstellung</i> von HULICO.	140
7.9	Vergleich der <i>Amerikanische-Einstellungen</i> von <i>RTL Aktuell</i> und HULICO.	141
7.10	Vergleich der <i>Halbnahe-Einstellung</i> und <i>Nahe-Einstellung</i> von <i>RTL Aktuell</i> und HULICO.	142
7.11	Fixpunkt während einer Zoomfahrt	143
7.12	Vergleich der Einstellungen bei unvorhergesehenen Änderungen an der Szene	144
7.13	Antwort-Möglichkeiten für Teil 1 des Fragebogens.	148
7.14	Deutsche Übersetzung der Fragen des <i>System Usability Scales</i> (SUS).	150
7.15	Auswertung Teil 1 des Fragebogens	157
7.16	Auswertung Teil 2 des Fragebogens	159
7.17	Abbildung des SUS-Scales in das amerikanische Notensystem von F bis A	160
A.1	Seite 1 des Evaluationsbogens	168
A.2	Seite 2 des Evaluationsbogens	169
A.3	Seite 3 des Evaluationsbogens	170
A.4	Seite 4 des Evaluationsbogens	171

ABBILDUNGSVERZEICHNIS

Kapitel 1

Einführung

In den letzten Jahren entschieden sich immer mehr Nachrichtenstudios für automatische Kamerasysteme. Diese tragen die Kamera während einer Sendung und sind in der Lage, zuvor gespeicherte Positionen immer wieder einzunehmen und definierte Fahrten auszuführen. Die aktuell bekanntesten Beispiele sind in den neuen Nachrichtenstudios der Sender ARD, ZDF und RTL installiert.

Die Systeme ermöglichen in Kombination mit Licht- und Videoeffekten neuartige, visuelle Erlebnisse für den Zuschauer. So können die Sendungen interessanter und moderner gestaltet werden. Außerdem wird durch die Reproduzierbarkeit der Kamerapositionen ein immer gleicher Bildeindruck vermittelt und so der Wiedererkennungseffekt [Boc11] des Formats¹ gefestigt.

Um eine Kameraposition auf die aktuelle Situation im Studio anzupassen, wird weiterhin ein Bediener benötigt. Dieser nimmt die Anweisungen der Regie entgegen und setzt sie am Kamerasystem um. Typischerweise beziehen sich diese Korrekturen auf eine geänderte Größe oder Position von Objekten im Bild. Der Bediener stellt die Schnittstelle zwischen dem Kamerasystem und der Regie dar und gibt dem System die nötige Flexibilität für Live-Sendungen.

Eine intelligente Steuerung für automatische Kamerasysteme muss sich, nach der in dieser Arbeit vertretenen Auffassung, wie ein menschengeführtes System in eine bestehende Produktionsumgebung einfinden und damit die nachfolgend aufgezählten Eigenschaften besitzen:

- Eine hohe Anwendungsflexibilität, um die unterschiedlichen Format-Definitionen verschiedener Sendungen abbilden zu können.

¹Ein Format ist „die spezifische kommerzielle Ausgestaltung und lizenzgebundene Festlegung einer Fernsehproduktion“ [Hic10]

1. EINFÜHRUNG

- Die Fähigkeit die im Studio gesprochene Sprache zu verstehen, um so gewohnte Anweisungsabläufe zu ermöglichen.
- Die Fähigkeit verbale Anweisungen in entsprechende Aktionen umzuwandeln sowie
- die Fähigkeit einen einmal erstellten Bildausschnitt immer wieder generieren zu können, unabhängig von Veränderungen an der Umgebung.

Der hier verfolgte Ansatz führt damit weg von starren achswert-basierten Positionsdefinitionen, die nur auf Knopfdruck angefahren werden können und hin zu einem System, das natürlich-sprachige Anweisungen verstehen und ausführen kann und Positionsdefinitionen durch bildbeschreibende Mittel speichert.

Diese Arbeit entstand im Rahmen der beruflichen Tätigkeit des Autors als Software-Entwickler bei der Firma Robotics Technology Leaders GmbH (RTLeaders) und basiert deshalb auf der von RTLeaders entwickelten Plattform RoboKam[©]. Die Ergebnisse der Arbeit sind jedoch von allgemeiner Natur und können auch für jedes andere System verwendet werden.

Im Folgenden wird eine Einführung in den Bereich der Nachrichten- und Magazinproduktion gegeben. Dabei wird auf Probleme hingewiesen, die sich aus dem Fehlen intelligenter Eigenschaften bei automatischen Kamerasystemen ergeben, um daraus dann das Thema zu entwickeln. Das Genre Nachrichten- und Magazinsendungen eignet sich besonders für den Einsatz einer intelligenten Kamerasteuerung, weil hier ein gleichbleibendes Erscheinungsbild der aufgenommenen Szene gewünscht ist. Durch formale Strukturen beschränkt sich die sonst kreative Arbeit an der Kamera auf die Einhaltung bestimmter Vorgaben.

1.1 Produktion von Nachrichtensendungen

Fernsehformate werden mit Hilfe sogenannter *Storybooks* geplant. Sie skizzieren den Verlauf der Sendung und die verschiedenen Kameraperspektiven. Die Realisierung dieser Skizzen geschieht durch den Regisseur in enger Zusammenarbeit mit einem Kameramann. Die hier festgelegten Einstellungen gelten für die regelmäßige Ausstrahlung der Sendung als verbindliche Vorlage.

Jeder Fernsehsender und jedes Format hat sein eigenes Erscheinungsbild. Dieses hängt nicht nur von der (oft virtuellen) Kulisse ab, sondern auch von der spezifischen Bildgestaltung (Kadrage) [Boc11] und der jeweiligen Kameraführung. Die Bildgestaltung bestimmt an welcher Position im Bild bestimmte Gegenstände oder Personen (z.B. Moderator) gezeigt werden bzw.

1.2 Problematik automatischer Kamerasysteme

welche Vorgaben diesbezüglich einzuhalten sind. Solche Vorgaben beziehen sich unter anderem auf gewünschte Größen und Positionen einzelner Objekte im Bild oder verbieten deren angeschnittene Darstellung.

Nachrichten entstehen über einen langen Prozess, der meist schon Stunden vor der Ausstrahlung beginnt. Aus einer Vielzahl von möglichen Meldungen müssen diejenigen ausgewählt werden, die zum Profil der Sendung passen und entsprechende Prägnanz besitzen [Fau08].

Abbildung 1.1 zeigt schematisch den Ablauf einer Nachrichtensendung. Auf die Eröffnungsfahrt (Opening) folgen die Begrüßung und der Themenüberblick. Darauf erfolgt nach redaktionellen Vorgaben die Präsentation der einzelnen Themen. Letztere werden oft just-in-time produziert, sodass die Reihenfolge erst im Rahmen der laufenden Sendung festgelegt werden kann. Aufgrund dieser dynamischen Struktur und dem fehlenden Sichtkontakt zwischen Regie und Kamera erfolgt die Kommunikation im Studio verbal, um so spontan agieren zu können. Nach der Verabschiedung erfolgt die Endfahrt (Ending).

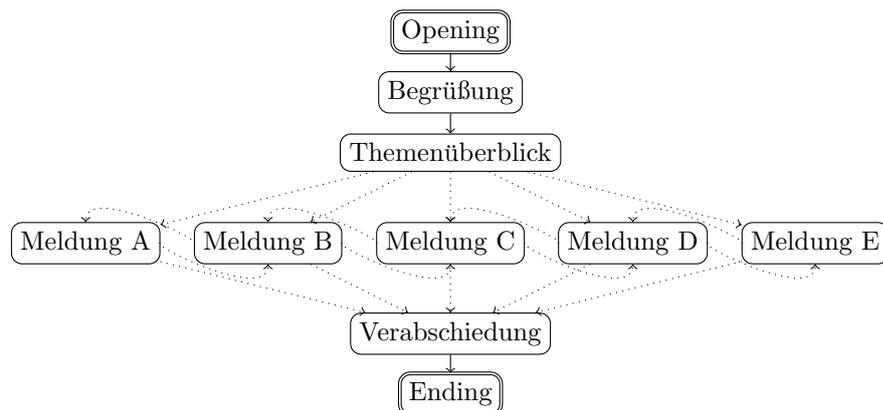


Abbildung 1.1: Schematischer Ablauf einer Nachrichtensendung. Auf die Eröffnungsfahrt (Opening) folgt die Begrüßung und der Themenüberblick. Es schließen sich die einzelnen Meldungen, nach Prägnanz und Verfügbarkeit geordnet, an. Nach den Meldungen erfolgt die Verabschiedung und evtl. eine Schlussfahrt (Ending).

1.2 Problematik automatischer Kamerasysteme

Kamerapositionen werden gegenwärtig entweder in Studio¹- oder in Systemkoordinaten² gespeichert und sind somit starr definiert. Unabsichtliche Änderungen an der aufzunehmenden Szene, durch z.B. einen anderen oder einen nicht richtig stehenden Moderator, können die

¹Studiokoordinaten beschreiben eine Position im Raum relativ zu einem Nullpunkt im Studio.

²Systemkoordinaten beschreiben eine Position relativ zum jeweiligen System (z.B. Achswerte des Systems).

1. EINFÜHRUNG

bisherigen automatischen Systeme nicht unter Beibehaltung des ursprünglichen Bildeindrucks kompensieren.

In Abbildung 1.2 nimmt ein automatisches Kamerasystem nach einer Kamerafahrt eine vorher gespeicherte Position im Raum wieder exakt ein und produziert damit in Bild (a) eine angeschnittene Moderatorin und somit ein “unharmonisch wirkendes Bild”.

Abbildung 1.2 (b) zeigt das vom Bediener manuell korrigierte Bild der Kamera. Dieser Vorgang wird im Rahmen der Arbeit als (Fehler-)Kompensation bezeichnet. Das Beispiel veranschaulicht die Notwendigkeit eines Bedieners, der die festgelegten Positionen auf die aktuelle Situation im Studio anpasst. Bisher verwendete, automatische Kamerasysteme erkennen weder Verfehlungen im Bild noch können sie verbal angewiesen werden, eine bestimmte Verfehlung zu beheben.



(a) fehlerhaft kadriert



(b) richtig kadriert

Abbildung 1.2: Gegenüberstellung einer fehlerhaft kadrierten mit einer richtig kadrierten Einstellungen aus *RTL Aktuell* vom 02.10.2010. In Bild (a) hat das Kamerasystem die gespeicherte Position exakt eingenommen. Die Moderatorin steht falsch, deshalb ist sie im Bild abgeschnitten abgebildet. In Bild (b) wurde die Fehlkadrierung manuell korrigiert. Bildquelle: <http://rtl-now.rtl.de>

Ein System mit der hier angestrebten intelligenten Steuerung hätte in diesem Fall entweder die Verfehlung von sich aus behoben und so selbstständig das Bild verbessern können oder es hätte angewiesen werden können, die Einstellung zu verbessern ohne dafür einen Operator bemühen zu müssen.

1.3 Ziele und Aufbau der Arbeit

In Zukunft werden automatische Kamerasysteme nicht einfach nur bestimmte Positionen einnehmen, sondern müssen in der Lage sein, ein bestimmtes Bild zu generieren und die verbal formulierten Anweisungen zu verstehen und auszuführen.

In der vorliegenden Arbeit werden Grundlagen für eine intelligente Steuerung eines automatischen Kamerasystems vorgestellt. Dazu gehören insbesondere die Identifikation von sendungsrelevanten Eigenschaften eines Bildes sowie die Definition eines vollständigen Satzes an Anweisungen zur verbalen Steuerung einer Kamera im Bereich der Nachrichten- und Magazinsendungen. Außerdem soll ein Bewegungsmuster identifiziert werden, das es erlaubt, gleiche Bildeindrücke für veränderte Studiobedingungen zu generieren. Die Ergebnisse dieser Arbeit sollen prototypisch umgesetzt und von Fachpersonal evaluiert werden.

Zunächst werden in Kapitel 2 der aktuelle Stand der Technik sowie verwandte Arbeiten in Bezug auf filmische und medizinische Kamerasysteme vorgestellt. Ergänzend werden einige aktuelle Arbeiten in Bezug auf sprachgeführte Systeme angeführt, bevor eine kurze Einführung in die kombinatorische Kategorial-Grammatik erfolgt.

Kapitel 3 zeigt wesentliche Aspekte der Kameraführung, die in der weiteren Arbeit berücksichtigt sind. Darunter sind neben theoretischen Aspekten der Kameraführung insbesondere solche enthalten, die in der Praxis Relevanz besitzen.

Für die intelligente Steuerung eines automatischen Kamerasystems stellt Kapitel 4, ausgehend von den in Kapitel 3 vorgestellten Aspekten, die nötigen Anforderungen zusammen.

Kapitel 5 stellt eine kombinatorische Kategorial-Grammatik vor, die in der Lage ist, die geforderten Anweisungen in eine logische Form zu übersetzen. Die Grammatik basiert vor allem auf den in den Kapiteln 3 und 4 gewonnenen Erkenntnissen.

Die Ausführungen von Kapitel 6 integrieren schließlich die im vorstehenden Kapitel dargestellte Sprachverarbeitung in ein Aktionsmodell, das die verwendeten Kommandos in praktische Aktionen umsetzt.

In Kapitel 7 wird ein Prototyp vorgestellt, der die entwickelten Konzepte aus den Kapiteln 5 und 6 konkret umsetzt. Dieser Prototyp dient als Grundlage für die Evaluation der hier präsentierten Ergebnisse. Dabei wurde die Leistungsfähigkeit des vorgestellten Systems anhand der in Kapitel 4 entwickelten Anforderungen beurteilt. Die neutrale Bewertung der Praxistauglichkeit erfolgte durch Regie-Mitarbeiter und erfahrene Kameraoperateure aus dem Nachrichten- und Magazinsbereich.

1. EINFÜHRUNG

Kapitel 2

Stand der Technik und wissenschaftlicher Überblick

Der Fokus der vorliegenden Arbeit liegt auf der Steuerbarkeit durch natürliche Sprache sowie auf einem intelligenten Verhalten eines Kamerasystems. Die Mensch-Maschine-Kommunikation in Zusammenhang mit filmischen Kamerasystemen sowie das studiotypische Verhalten einer Kamera ist ein bisher nicht untersuchter Anwendungsaspekt. Es werden deshalb Arbeiten vorgestellt, die thematisch im Bereich Einstellungsfindung, Fahrtbeschreibung von Kamerasystemen sowie im Bereich der natürlichsprachlichen Interaktion mit Systemen liegen.

In Kapitel 2.1 wird der Begriff Kamerasystem allgemein sowie die übliche Arbeitsweise mit einem solchen System eingeführt. In Kapitel 2.2 werden Veröffentlichungen vorgestellt, die sich auf die Einstellungsfindung beziehen. Außerdem werden die Begriffe Kameraplanung und Kinematographie eingeführt. In Kapitel 2.3 werden Publikationen vorgestellt, die Fahrten von Kameras thematisieren. Medizinische Kamerasysteme sowie Arbeiten in diesem Bereich, werden in Kapitel 2.4 vorgestellt. Das Kapitel schließt mit einigen Veröffentlichungen aus dem Bereich der sprachgeführten Systeme in Kapitel 2.5 sowie einer Einführung in die kombinatorische Kategorial-Grammatik in Kapitel 2.6.

2.1 Kamerasysteme beim Film und Fernsehen

Unter einem automatischen Kamerasystem wird im Folgenden eine kameraführende Kinematik verstanden. Diese kann über einen Joystick ferngesteuert werden und ist in der Lage zuvor gespeicherte Positionen einzunehmen sowie Fahrten auszuführen. Wir fordern hier für ein automatisches Kamerasystem, dass mindestens die Achsen Pan (horizontale Orientierung), Tilt

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

(vertikale Orientierung), Zoom und Fokus angesteuert werden können. In dieser Arbeit liegt der Schwerpunkt jedoch auf speziellen Kamerasystemen, die es erlauben, neben oben genannten Achsen, auch die Position (X-, Y- und Z-Achse) einer Kamera zu verändern. Solche Kamerasysteme sind in der Lage unterschiedliche Perspektiven für den gleichen Bildausschnitt zu generieren. Abbildung 2.1 (a) zeigt die Camerobot RoboKam[©]. Sie liegt dieser Arbeit zugrunde.

Die Vorteile von automatischen Kamerasystemen liegen in der Generierung neuer Perspektiven, in größeren Reichweiten und Höhen sowie in geringeren Personalkosten. Die hochgenaue Reproduzierbarkeit der einzelnen Fahrten ist bei Formaten mit vordefiniertem Ablauf, wie beispielsweise bei Nachrichtensendungen wichtig, um visuelle Effekte speziell auf die Kamerabewegung abstimmen zu können. Die Schwenks auf die jeweiligen Videobeiträge in der *ZDF heute*-Sendung sind Beispiele hierfür.



(a) Camerobot RoboKam[©]



(b) Shotoku-Panel

Abbildung 2.1: (a) Kamerasystem RoboKam7 von RTLeaders GmbH (b) Bedienpaneel (Joystick-Konsole) der Firma Shotoku. Der im Rahmen dieser Arbeit entwickelte Prototyp basiert auf dem in (a) dargestellten automatischen Kamerasystem. Das Shotoku-Bedienpaneel ist eine typische Form der Steuerung eines automatischen Kamerasystems.

Bildquellen: <http://www.rtleaders.com> und <http://www.shotoku.co.uk>.

Die Arbeitsweise mit automatischen Kamerasystemen ist weithin gleich. Es werden mit der Kamera bestimmte Studio-Positionen angefahren, die Kamera entsprechend eingestellt und diese Konfiguration dann gespeichert. Soll eine Fahrt durchgeführt werden, so wird diese durch eine Start-, optional mehreren Zwischen- und eine Endposition bestimmt. Die jeweiligen Übergänge zwischen diesen definierten Positionen werden durch Interpolation der einzelnen Achsen berechnet. Es gibt zwei verschiedene Arbeitsweisen für automatische Kamerasysteme: die manuelle Kameraführung und die Programmierung über Schlüsselpositionen bzw. über Bedingungen.

2.1.1 Manuelle Kameraführung

Alle automatischen Kamerasysteme beherrschen mindestens die manuelle Kameraführung. Unter manueller Kameraführung verstehen wir hier die direkte Interaktion des Operators (Kameramann) mit dem System. Die Kamera wird dabei durch einen Bediener mit Hilfe eines Joysticks geführt. Der Joystick kann in der Software-Oberfläche oder über ein Bedienpaneel zur Verfügung stehen. Abbildung 2.1 (b) zeigt ein Bedienpaneel der Firma Shotoku.

2.1.2 Programmierbare Kameraführung

Eine Kamerabewegung wird über Schlüsselstellen (Stützstellen) definiert. Meist sind das Start- und Endposition sowie eine oder mehrere Zwischenpositionen. Zwischen diesen diskreten Positionen wird interpoliert und so eine flüssige Bewegung erzeugt. Der Vorteil dieser Technik liegt in der intuitiven Verständlichkeit des Programmierprozesses. Prinzipiell kann jede Fahrt über die entsprechende Anzahl an Stützstellen programmiert werden. Der Nachteil liegt darin, dass gerade bei Radial-Bewegungen um die Vertikal-Achse des Systems das betrachtete Objekt im Fokus schnell zu *schwimmen* beginnt. Der Begriff *schwimmen* bezeichnet eine ungewollte horizontale Bewegung eines Objekts im Bild während einer Fahrt (siehe Kapitel 3.1.5).

Eine weitere Möglichkeit der Programmierung liegt darin, einer Fahrt bestimmte Parameter (Bedingungen) zu Grunde zu legen. Solche Parameter können sich auf die Bewegungsgeschwindigkeit der Kamera, deren Beschleunigung oder auf die Ausgestaltung der Bahn selbst beziehen. Für den Verlauf der Bahn sind unterschiedliche Modelle, wie z.B. Kreisbahnen, Spiralbahnen oder Wellenbahnen möglich. Wird beispielsweise eine Kreisfahrt um ein bestimmtes Objekt herum gewünscht, kann das durch die Positionsinformation des Startpunkts, des Endpunkts sowie des Ziels, auf das die Kamera ausgerichtet werden soll, realisiert werden. Der Operator, muss in diesem Fall nur die Position der Kamera festlegen, während die Orientierung durch das System berechnet wird. Die Parameter bestehen dann aus dem Zielpunkt und den Radien, jeweils zu Beginn und am Ende der Fahrt. Aus diesen Parametern kann so eine homogene Kreisfahrt interpoliert werden, die ein *schwimmen* des Objekts im Bild vermeidet. Diese Programmier-Technik erlaubt es relativ schnell eine gewünschte Fahrt zu realisieren, weil eine zeitraubende Korrektur der Kameraorientierung nicht notwendig ist. Allerdings ist bei diesem Vorgehen ein mathematisches Verständnis für die Wahl des zur gewünschten Fahrt passenden Modells nötig und das Kamerasystem muss ein Programmiermodell bieten, das die geforderte Bedingung erfüllt. Die Programmierung durch Bedingungen stellt einen Spezialfall dar und wird deshalb meist nur im Filmbereich oder für bestimmte Typen von Werbeaufnahmen gebraucht.

Intelligente Funktionen sowie Sprachsteuerung gibt es für Kamerasysteme im Film- und Fernserebereich bisher nicht.

2.2 Arbeiten mit Bezug auf Kamera-Einstellungen

Die ersten Arbeiten, die bestimmte Einstellungen einer Kamera fokussieren, stammen aus der Entwicklung der virtuellen 3D-Welten in den 80er Jahren. Deshalb werden im Folgenden einige grundlegende Arbeiten aus dieser Zeit angeführt. Typische Begriffe in dieser Domäne sind Kameraplanung und Kinematographie.

Kameraplanung ist ein Begriff aus dem Bereich der 3D-Computergrafik. Diese Domäne steht der Film- und Fernsehwelt sehr nah, weil viele Problemstellungen, die bei der Verwendung realer Kameras auftreten, auch in der virtuellen 3D-Welt zu bewältigen sind. Die Kameraplanung thematisiert die Positionsfindung der Kamera, jedoch steht hier, neben der Interaktion mit dem 3D-Modell vor allem die Okklusionsvermeidung im Vordergrund (s. [GW92, Hel04]).

Die Kinematografie ist der Äquivalenzbegriff zur Kameraplanung aus dem Film-Bereich und beschäftigt sich mit der *Positions-Findung* am Filmset. Eine vorgegebene Szene und die beschriebene Stimmung in dieser Szene soll durch eine optimale Kamerapositionierung und -führung bestmöglich dem Zuschauer präsentiert werden. Die Kameraposition wird dabei so gewählt, dass der Zuschauer genug Information sehen kann, um die Situation im Ganzen zu erfassen. Irritationen des Zuschauers z.B. durch wechselnde Bildkompositionen, werden vermieden. Für einen Kinematographen ist die Bewegung der einzelnen Akteure, die Beziehungen dieser untereinander und die Emotionen der einzelnen Charaktere bei der Kameraplanung zu berücksichtigen. Frühes Ziel der Kinematographen war es, den besten Platz für die Kamera zu finden. Dieser Platz entsprach dem „idealen Platz des Theaterbesuchers“ [HA05].

Die vermutlich erste wissenschaftliche Arbeit im Bereich der Kameraplanung stammt von J. Blinn aus dem Jahr 1988 [Bli88]. Blinn berechnete Kamera-Positionen in einer Szene, die aus einem Raumschiff und einem Planeten bestand. Dabei sollte die Kamera so positioniert werden, dass beide Objekte an bestimmten Positionen im Bild dargestellt wurden. Der damalige Ansatz war rein geometrischer Natur und nur für ein Loch-Kameramodell [HZ04] mit genau zwei Objekten verwendbar.

Drucker [Dru94] entwickelte ein System, das auf Bedingungen basiert. Drucker stellt vier Beispielszenarien für die Positionierung einer Kamera vor: Die Erforschung einer neuen Umgebung, die Planung von Vorgehensweisen, die Ausbildung und die Kontrolle über eine virtuelle

2.2 Arbeiten mit Bezug auf Kamera-Einstellungen

Welt. Die Kameraposition wird dabei über Zielfunktionen beschrieben. Die Lösung der jeweiligen Aufgabe geschieht über sequentielle quadratische Programmierung. Da diese Methode die Gefahr birgt in lokalen Minima zu stagnieren, formuliert Drucker Ratschläge, die bei der Verwendung des Systems berücksichtigt werden sollten: Das Problem sollte so formuliert werden, dass kein lokales Minimum beschrieben wird. Der Arbeitsraum sollte diskretisiert werden und es sollte mit einer Start-Konfiguration begonnen werden, die nah an der Zielkonfiguration ist. Außerdem sollte es dem Benutzer möglich sein, in die Berechnung einzugreifen.

Bares et al. [BTM00] findet die beste Kameraposition durch einen iterativen Ansatz. Hier werden für eine Menge von diskreten Kamerapositionen die gegebenen Bedingungen (Constraints) auf Erfüllung getestet. Über die Zielfunktion $\text{satisfaction} = \sum (\text{prio}_i \cdot \text{satisf}_i)$ wird die bestmögliche Kameraposition bestimmt. Es werden Bedingungen, wie `look_at_point`, `obj_in_field_of_view`, `cam_pos_in_region` und einige mehr implementiert. Mit diesem Ansatz kann man für beliebig viele Objekte im Bild die beste Kameraposition bestimmen. Allerdings ist der iterative Ansatz sehr berechnungsintensiv und einzelne Constraints können andere obsolet bzw. unmöglich machen.

CAMPLAN [HO00] von Halper und Olivier findet Kamerapositionen anhand von übergebenen Bildeigenschaften. Die Bildeigenschaften resultieren aus Positions-, Orientierungs-, Größen- und Sichtbarkeitseigenschaften für bestimmte Objekte im Bild. Die Attribute können sowohl absolut, wie auch relativ zu anderen Objekten angegeben werden. Die Positionsbestimmung wird durch einen genetischen Algorithmus durchgeführt. Genetische Algorithmen entsprechen dem Vorgehen der Evolution. Mit jedem Evolutionsschritt setzen sich gute Gene durch. Schlechte Gene werden dagegen nicht weiter berücksichtigt. Bei genetischen Algorithmen wird der zufällig gewählte Ausgangszustand binär beschrieben und als *Chromosom* bezeichnet. Mit jeder Iteration werden im Chromosom jeweils die besten Gene weiter vererbt. Nachteil dieser Art von Algorithmen ist, dass die Anzahl der nötigen Evolutionsschritte im Vorhinein nicht bestimmt werden kann und damit die Berechnungsdauer mitunter sehr lang ist. Das System benötigt unter Verwendung aller sieben Freiheitsgrade der Kamera zwischen Minuten und Stunden, um eine bestimmte Position zu finden. Zu den sieben Freiheitsgraden einer Kamera zählen drei Freiheitsgrade der Position, drei für die Orientierung und ein Freiheitsgrad für die Kontrolle des Zooms.

Pickering [Pic02] erweitert diesen Ansatz und stellt eine Image Description Language (IDL) vor, mit der die Kameraposition durch Lösung eines Constraint-Optimization-Problem (COP) berechnet wird. Das COP wird ebenfalls mit Hilfe eines genetischen Algorithmus gelöst. Eine

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

interaktive Verwendung dieses Ansatzes ist nicht möglich. Es werden fünf Szenarien behandelt: Eine Portrait-Aufnahme, eine Over-The-Shoulder-Aufnahme, eine Action-Aufnahme aus einer Kochsendung, eine Landschaftsaufnahme, die ein öffentliches Gebäude zeigt und eine Orientierungsaufnahme, in der eine Waschküche dargestellt ist. Anhand dieser Szenarien wird die IDL entwickelt um damit allgemein Aufnahmen beschreiben zu können. Ein Beispiel für diese formale Sprache lautet:

```
AND( SHOW(Romeo),
      SHOW(Juliet),
      THREE_QUARTER_SHOT(Romeo),
      THREE_QUARTER_SHOT(Juliet)
    )
```

Diese Sequenz sagt aus, dass sowohl *Romeo* als auch *Juliet* in einer Einstellung gezeigt werden sollen, in der beide schräg an der Kamera vorbei schauen. Luthard [Lut11] nimmt die constraintbasierte 3D-Kamerapositionierung wieder auf und vergleicht die Optimierungsalgorithmen inkrementelle Brute-Force-Methode (IBF), Simulierte Abkühlung (SA) und Kuckuck Suche (KS). Dabei zeigt sich, dass der IBF-Ansatz „deutliche Schwachstellen in der Laufzeit“ [Lut11] hervorbringt. Für die Algorithmen SA und KS wird festgestellt, dass die Laufzeiten stark variieren. Dennoch ist nach [Lut11] für die Berechnung von Kamerapositionen „die SA der CS vorzuziehen“.

Tomlinson et al. [TBN00] haben ein Multiplayer-Computerspiel entwickelt, bei dem das Licht in der Szene und die Kamera durch einen Agenten gesteuert wird. Abbildung 2.2 zeigt eine Szene aus diesem Spiel. Der Zustand der Avatare wird durch *Emotionen* definiert. Es wird zwischen glücklich, traurig, wütend, überrascht, ängstlich und angewidert unterschieden. Die Emotion einer Figur bestimmt, ob sie von der Kamera bevorzugt gezeigt wird. Die Art der Darstellung entscheiden die Figuren anhand einer definierten Motivation. Motivationen können *DesireForTwoShot*, *DesireForEstablish* oder *DesireForCloseUp* sein. Bei *DesireForTwoShots* wünscht die Figur eine *Zweier-Einstellung*¹, bei *DesireForEstablish* wird eine *Totale-Einstellung*¹ gewünscht und bei *DesireForCloseUp* ein *Nahe-Einstellung*¹. Der Kamera-Agent kombiniert nun Sensorinformationen mit Motivation und Emotion und ermittelt daraus in einem Aktions-Selektions-Mechanismus, welche Aktion er als nächstes ausführt. Die automatische Kameraführung unterstützt die Spieler beim Steuern der jeweiligen Spielfigur. In der vorliegenden Arbeit liegt der Fokus jedoch auf einer durch den Menschen gesteuerten Kamera.

¹Die einzelnen Einstellungen werden in Kapitel 3.1.2 beschrieben



Abbildung 2.2: Abbildung des Multiplayer-Computerspiels von Tomlinson et al. Avatare können hier durch unterschiedliche Motivationen die Aktionsselektion der Kamera beeinflussen.

Das System von Tsuda et al. [THYK03] versucht möglichst autonom Kameraeinstellungen in einer bekannten Szene einzunehmen. Dies wird beispielhaft an einer Interview-Szene mit Flipchart-Erklärteil vorgestellt. Über ein Personen-Tracking-System wird die Position der Moderatoren festgelegt. Mit Hilfe dieser Information wird geschätzt, wo sich die Köpfe der Moderatoren befinden und wie die Kamerasysteme ausgerichtet werden müssen. Als Kamerasysteme kommen hier drei Stative mit den automatisch kontrollierbaren Achsen Pan, Tilt, Zoom und Fokus zum Einsatz. Die Szene wird über ein Subject-Detection-System erkannt. Die beteiligten Personen werden durch LEDs auf dem Rücken markiert, anhand derer der Standort jeder einzelnen Personen berechnet werden kann. Die Einstellungsgröße wird durch die relative Größe des Kopfes im aufgenommenen Bild definiert. Die Größe des abgebildeten Kopfes wird durch Farbdetektion erkannt. Die drei Kameras kommunizieren untereinander so, dass immer mindestens eine Kamera „onAir“ ist. Diese Kamera zeigt ein statisches Bild und agiert nicht mit. Das ganze System basiert auf der Triangulation der Szene. Die entstehenden Bilder sind fest definiert und stark vom Szenen-Setup abhängig. Wird eine Kamera umpositioniert, müssen alle Berechnungen neu parametrisiert werden. Der in der vorliegenden Arbeit vertretene Ansatz ist insofern verwirklicht, als dass feste Einstellungen existieren. Eine verbale Interaktion findet jedoch nicht statt.

2.3 Arbeiten mit Bezug auf Kamerafahrten

Die bisher vorgestellten Arbeiten behandeln die Positionsbestimmung der Kamera. Die folgenden Arbeiten stellen die Bewegung der Kamera in den Vordergrund.

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

Das CINEMA-System von Drucker [DGZ92] dient der Beschreibung von Kamerabewegungen. Es arbeitet in einer virtuellen Welt, in der die Position und die Sichtbarkeit aller Objekte abfragbar ist. Primitive Hilfs-Funktionen erlauben es, Positionen der einzelnen Objekte innerhalb der Szene zu ändern sowie den Zustand und die Orientierung der Kamera zu erfassen. Mit Hilfe dieser Grundfunktionen erfolgt dann eine Beschreibung von Kamerafahrten. CINEMA besteht im Wesentlichen aus einer Datenbank und einem Parser. Die Datenbank stellt Informationen (auch im zeitlichen Verlauf) über alle Objekte zur Verfügung. Der Parser interpretiert diese und führt sie dann aus. Abbildung 2.3 zeigt eine CINEMA-Beschreibung des Vertigo-Effekts¹.

```
proc vertigo_shot(obj, rate, no_frames) {
  // get the angle subtend by the object
  angle := get_angle_height(obj);
  // get the cameras field of view
  fov := cam_fov();
  // compute the percentage of the fov
  // which the object subtends
  percent := angle/fov;
  for (i=0;i<no_frames; i++) {
    // truck in the specified direction
    //at the specified rate
    cam_truck(rate);
    // set the field of view so that the
    // object subtends the same percentage
    // as before
    frame_it(obj, percent);
    render();
  }
}
```

Abbildung 2.3: Formale Beschreibung des Vertigo-Effekts in CINEMA.

Vorteil eines solchen high-level Zugangs ist laut Drucker [DGZ92], dass mögliche Rausch-Inputs, z.B. durch Joysticksteuerung mithilfe von High-Level-Beschreibungen umgangen werden können. Bewegungen sind leicht reproduzierbar. Nachteilig, ist, dass Prozeduren nicht kombiniert werden und dass keine Bedingungen angegeben werden können. Für die Verwendung im

¹Beim Vertigo-Effekt wird die Kamera auf ein Objekt zubewegt. Dabei kompensiert der Zoom die Größenänderung des Objekts. So bleibt das Objekt im Bild klar zu erkennen, während sich der Hintergrund ändert. Zuerst eingesetzt in A. Hitchcocks Film: Vertigo - Aus dem Reich der Toten 1958

Rahmen einer intelligenten Kamerasteuerung im Nachrichtenstudio ist dieser Ansatz zu formal. Normales Studiopersonal hat im Allgemeinen nicht die Programmiererfahrung, um auf diese Art und Weise eine Kamerafahrt zu programmieren.

Ein weiteres System mit Bezug auf Kamerafahrten ist ShowMotion. Dieses System ordnet sich in den Bereich der 3D-Modeling-Tools ein. Abbildung 2.4 zeigt die Möglichkeiten des Systems. Es werden 3D-Modelle generiert und bei Bedarf präsentiert. Die Präsentation solcher 3D-Modelle geschieht nach Burtnyk et al. [BKFK06] im Allgemeinen durch gespeicherte Positionen der virtuellen Kamera. In dieser Arbeit wird die statische Präsentationsform durch eine bewegte Kamera vollzogen. Üblicherweise werden bewegte Szenen in einem CAD-Programm entweder durch eine Drehscheibe bei feststehender Kamera oder durch Übergänge zwischen gespeicherten Positionen (Bookmarks) durchgeführt. Eine mit einer Drehscheibe aufgenommene Szene wirkt nach [BKFK06] aufgrund immer wiederkehrender Information schnell langweilig und Bookmarks erzeugen oft zu wenig Bewegung an uninteressanten und zu viel an interessanten Stellen. Das Ziel von ShowMotion ist es, reproduzierbare Eindrücke auf ein 3D-Modell zu generieren. Im dort vorgestellten Beispiel wird so eine bessere Vergleichbarkeit zwischen den aktuellen Auto-Modellen und denen des Vorjahres generiert.

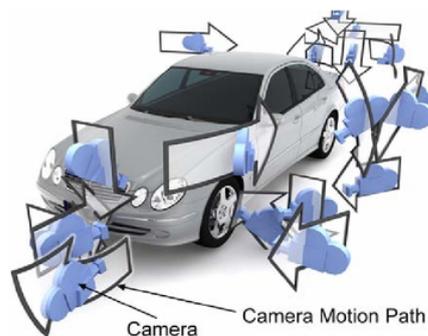


Abbildung 2.4: ShowMotion von Burtnyk et al. Das System bietet die Möglichkeit Aspekte eines 3D-CAD-Modell ansprechend zu Präsentieren. Dafür werden für die jeweiligen Aspekte entsprechende Fahrten aus einer Datenbank geladen.

ShowMotion dient der Planung von Kamerabewegungen im 3D-ShowCase [BKFK06]. Mit diesem Werkzeug können 3D-Modelle ansprechend präsentiert werden. Außerdem bietet es die Möglichkeit vordefinierte Fahrten oder stehende Bilder zu generieren. Nach einem Klick auf das jeweilige interessante Teil, wird die Datenbank durchsucht, um zum Klickpunkt passende Bilder zu präsentieren. Diese Arbeit zeigt die Verwandtschaftsbeziehung zwischen 3D-Welten

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

und realem Fernsehen. Allerdings liegt bei ShowMotion der Fokus auf der Auswahl einer bestimmten Fahrt, während das Ziel der vorliegenden Arbeit auf einer intelligenten Kamera liegt, die bestimmte Anweisungen ausführen kann.

2.4 Kamerasysteme in medizinischen Umgebungen

Neben den filmischen, kameratragenden Kinematiken gibt es medizinische bzw. operations-unterstützende Roboter, die ähnliche Anforderungen wie ein kameratragendes System am Filmset haben: Sie visualisieren eine bestimmte Szene. Im praktischen Einsatz befinden sich hier beispielsweise die Systeme *EndoAssist*, *ZEUS* und *da Vinci* (vgl. Abbildung 2.5). Diese Geräte tragen sowohl die für die Operation nötigen Instrumente, wie auch eine Kamera, mit der die Operationsfläche im inneren des Körpers beobachtet werden kann. Der *EndoAssist* wird durch die Kopfbewegung des Operators gesteuert. Das *Da Vinci*-System wird über eine externe Fernsteuereinheit kontrolliert und *Zeus* [MR03] kann sowohl verbal wie auch konventionell gesteuert werden. Die Vorteile dieser Systeme liegen in der hohen Präzision, der ermüdungsfreien Ausführung der jeweiligen Aufgabe sowie in der Möglichkeit die Operationsfläche dreidimensional darzustellen [KKKL05, CEC⁺05].



Abbildung 2.5: Medizinische kameratragende Systeme. Diese Systeme tragen ebenfalls eine Kamera und sind vergleichbar mit automatischen Kamerasystemen. Allerdings liegt hier der Fokus auf dem ruhigen Tragen der Kamera.

Krupa et al. [KGD⁺03] stellen ein visuell gesteuertes Operationssystem vor, bei dem die Position des jeweiligen Instruments relativ zur Umgebung erkannt wird. So sollen auch dann verletzungsfreie Bewegungen des Instruments möglich sein, wenn sich das Instrument selbst nicht im Sichtbereich der Kamera befindet. Anhand von mehreren, durch Laser-Projektionen

erstellten, optischen Marken auf dem zu behandelnden Organ kann die Position des Instruments relativ zur Kamera berechnet werden. So kann das Instrument sicher in den Blickbereich der Kamera zurückgeführt werden. Zur Positionierung des Instruments wird ein Algorithmus verwendet, der anhand der Pixel-Koordinaten der Laserpunkte auf dem Organ den Abstand zwischen Organ und Instrument berechnet. Die vorgestellte Bildverarbeitung erlaubt es, die erkannten Marker durch Software zu beseitigen, sodass dem Operateur ein weitgehend unmodifiziertes visuelles Feedback zur Verfügung steht.

Fowler et al. [FHN⁺09] adressieren die geringe Aussagekraft von zweidimensionalen Bildern, wie sie mit den bisherigen Systemen generiert werden. In dieser Machbarkeitsstudie wird eine verbesserte Darstellung für minimal-invasive Operationen vorgestellt. Die Arbeit präsentiert eine stereoskopische Kamera zur Generierung eines dreidimensionalen Bildes. Sie ist fernsteuerbar und enthält einen digitalen Zoom sowie Tracking-Technologien, die es der Kamera erlauben weitgehend autonom zu agieren.

Osa et al. [OSK10] stellen ein Operationssystem vor, das autonom die Kamera einer laparoskopischen Operation steuern soll. Laparoskopische Eingriffe sind eine Untergruppe minimalinvasiver Eingriffe. Sie werden mit Hilfe einer Kamera durch kleine Öffnungen (Ports) im Inneren des Körpers durchgeführt. In dieser Arbeit wird insbesondere die Bewegung des Instruments an der Eintrittsstelle in den Körper berücksichtigt. Das System besteht aus vier Robotern, die an der Decke über dem Operationstisch montiert sind. Drei dieser Roboter tragen medizinische Instrumente während ein weiterer Roboter die Kamera führt. Die Steuerung der Roboter geschieht durch bildbasiertes *Visual-Servoing*. Dabei wird der Roboter auf der Basis eines Videobildes bewegt. Die Arbeit demonstriert, dass Visual-Servoing die Instrumente auf unter 1mm Genauigkeit positionieren kann. Außerdem wird durch das vorgestellte Berechnungsmodell die Bewegung der Kamera an der Eintrittsstelle in den Körper des Patienten auf weniger als 1mm beschränkt.

Nach King et al. [KRP⁺13] lenken die fortlaufenden Anweisungen eines Operateurs an einen menschlichen oder automatischen Kameraträger den Operateur unnötig von seiner Aufgabe ab. Sie entwickelten deshalb eine Test-Plattform für eine autonome Kamerakontrolle, die die verwendeten Instrumente optisch verfolgt. Die Kamera wird somit automatisch und ohne menschliche Interaktion positioniert. Das System erkennt die verwendeten Instrumente im übertragenen Bild und verfolgt diese mit der Kamera.

In der Arbeit von Buchs et al. [BVF⁺13] werden die in einer minimal-invasiven Operation verwendeten Instrumente durch optische Tracking-Marken ergänzt und mit der Kamera verfolgt.

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

So wird die dreidimensionale Darstellung der, in jener Arbeit zu operierenden, Leber um die Darstellung der Instrumente erweitert. Ein Ziel-System erlaubt es, den Abstand zwischen Tumor und Instrumenten-Spitze darzustellen, um eine sichere Schneide-Grenze einzublenden. Für die Arbeit wird das Da-Vinci-System verwendet (vgl. Abbildung 2.5 (b)).

In dem Artikel von Kim et al. [KLK⁺04] wird ebenfalls ein Operations-Assistenz-System für laparoskopische Eingriffe präsentiert. Es stellt die Schnittstelle zwischen einem medizinischen Kamerasystem und einem sprachgeführten System dar. Die Führung der Kamera soll nach [KLK⁺04] nicht mehr durch einen Menschen, sondern durch ein Assistenz-System geschehen, weil ein solches keine Erschütterungen durch die Handführung ausführt, nicht müde wird und den Chirurgen weniger behindert. Das System kann sich in zwei Richtungen verformen und außerdem translatorisch ausgefahren werden. Anhand eines intelligenten Tracking-Algorithmus, der die Bewegungen der Instrumente verfolgt, erkennt das System die jeweiligen Operationschritte und kann so immer die bestmögliche Ausrichtung der Kamera sicherstellen. Um den Operateur bestmöglich zu unterstützen, sind zwei Modi implementiert: Ein Automatikmodus und ein Modus mit verbaler Steuerung. Das System interpretiert dann Kommandos, wie *Up*, *Down*, *Left*, *Right*, *Zoom-in* und *Zoom-out*. Die Umschaltung der Modi erfolgt ebenfalls verbal durch „Track“ bzw. „Voice“.

Alle diese vorgestellten Arbeiten zeigen, dass medizinische, kameraführende Systeme durchaus Ähnlichkeiten zu filmischen kameraführenden Systemen haben, denn auch diese zeigen eine bestimmte Szene – das Operationsfeld – bestmöglich. Allerdings steht in den vorgestellten Arbeiten immer die Visualisierung bzw. die exakte Positionierung der Instrumente im Vordergrund. Die Kommandos zum Steuern der hier vorgestellten Systeme sind sehr einfach aufgebaut und nicht natürlichsprachlich formuliert. In der vorliegenden Arbeit wird ein System jedoch nur dann als intelligent bezeichnet, wenn mit ihm genauso kommuniziert werden kann, wie mit einem menschlichen Arbeitspartner.

2.5 Sprachgeführte Systeme

Intelligente Maschinen müssen sich am Menschen messen lassen. Erst wenn der Mensch nicht mehr unterscheiden kann, ob er mit einem realen Menschen interagiert oder mit einer Maschine, kann nach A. M. Turing [Tur50] eine Maschine als intelligent bezeichnet werden. Bei direkter Interaktion mit der Maschine ist die natürliche Sprache ein wichtiges Mittel auf diesem Weg.

Im Folgenden werden einige Arbeiten und Systeme vorgestellt, die durch natürliche Sprache steuerbar sind.

Das System von Kim et al. [KSJ⁺10] ist ein Roboter-System, das automatisch Bilder aufnehmen kann. Es folgt dabei empfangenen Stimm-Geräuschen. Wenn in der Richtung, in der eine Stimme detektiert wurde, ein Mensch anhand einer Farb-Segmentation erkannt wurde, wird dieser fotografiert. Das Bild wird nach der *Zweidrittel-Regel*¹ kadriert, die besagt, dass Objekte, die jeweils auf einem Drittel des Bildes positioniert sind, besonders ästhetisch wirken. Dieses System entspricht im weitesten Sinn einer sprachgeführten Kamera und ist deshalb zu Beginn des Kapitels aufgeführt. Es versteht jedoch keine gesprochenen Kommandos.

Holada und Pelc [HP08] stellen ein universelles, lernendes *voice-control system* für die Roboter-Interaktion vor, bei dem die Szene durch eine fest montierte Kamera aufgenommen wird. Mit Hilfe von Sprachanweisungen kann der Roboter in einem fest definierten Szenario bewegt und neue Abläufe in Form von Makros eingelernt werden. Das System verwendet eine proprietäre Spracherkennung [Nou00], die auf vier Wort-Gruppen aufbaut. Eine Gruppe enthält Fahrplananweisungen wie *move up* oder *stop*. Die zweite Gruppe enthält alle nutzbaren, aber noch nicht verwendeten Wörter. Die dritte Gruppe enthält Bezeichner für Makros in Form von Wörtern und kurzen Sätzen. Die vierte Gruppe enthält *build-in*-Anweisungen wie *robot calibration* oder *begin learning* etc. Das System ermöglicht die verbale Kontrolle eines Roboters im Test-System und erlaubt Anweisungen in der Form:

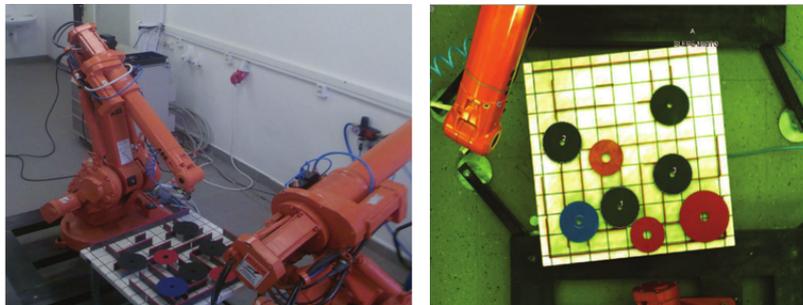
```
Operator: Search black disks
System: I am searching ... Four disks were found
Operator: Move on first
System: I am moving ... Done
Operator: Take it
```

Die Abbildungen 2.6 (a) und (b) zeigen den für diese Arbeit verwendeten Aufbau sowie die Sicht der Kamera. Jene Arbeit hat die Anweisung des Roboters im Fokus und nicht die Arbeit mit einer Kamera. Sie enthält folglich keine bildbeschreibenden Aussagen und verwendet keine Grammatik zur Interpretation der Anweisungen.

Aktuell stellt die Interpretation von Indoor-Navigationsanweisungen einen Schwerpunkt für die Verarbeitung natürlicher Sprache dar. In diesem Themengebiet werden natürlichsprachliche Weganweisungen erst in eine formale Sprache und schließlich in einen Pfad auf einem Gebäudeplan

¹Die Zweidrittel-Regel ist eine übliche Vereinfachung des Goldenen Schnitts. Der Goldene Schnitt wird als besonders ästhetische Bildaufteilung angesehen [Fry10, Wag14, Fei01].

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK



(a) Bild der Roboter-Szene aus [HP08]

(b) Sicht der Kamera aus [HP08]

Abbildung 2.6: Voice-Control-System von Holada und Pelc. Mit Hilfe von Sprachkommandos kann der Roboter in einem Industrie-Szenario bewegt werden. Eine Kamera bietet ein statisches Bild der Szene. Durch Makros werden neue Abläufe gelernt.

übersetzt. Die im Folgenden beschriebenen Arbeiten von Wong und Mooney [WM06], Matuszek et al. [MFK10], Chen und Mooney [CM11] und von Matuszek et al. [MHZF12] lassen sich diesem Themengebiet zuordnen.

In der Arbeit von Wong und Mooney [WM06] wird eine synchrone kontextfreie Grammatik als natürlichsprachliche Schnittstelle verwendet. Diese Grammatik erlaubt es einen in einer Sprache generierten Satz in eine formale Sprache zu übersetzen. Dies wird erreicht, indem jedes Non-Terminal auf der linken Seite der Produktionsregeln auf ein Paar von Anweisungen $\langle \alpha, \beta \rangle$ abgebildet wird. Dabei sind die Non-Terminals von α eine Permutation der Non-Terminals von β . Die synchrone kontextfreie Grammatik wird für die Übersetzung aus einer natürlichen Sprache in eine formale Beschreibungssprache verwendet. Bei diesem Ansatz gibt es für einen gegebenen Satz e oft mehrere Ableitungen. Es muss folglich ein Mechanismus gefunden werden, der die richtige Ableitung auswählt. Wong und Mooney verwenden die Reihenfolge der einzelnen Worte als Zuweisungskriterium.

Matuszek et al. [MFK10] verwenden *statistical machine translation* (SMT) um natürliche Sprache in eine formale Sprache zu überführen. Die SMT basiert ebenfalls auf einer synchronen kontextfreien Grammatik und verwendet Methoden des *Machine Learning*, um natürliche Sprache zu interpretieren. Die Arbeit basiert auf automatisch generierten Karten, die keine Markierung bestimmter Bereiche oder sonstige Orientierungsmarken enthalten. Die betrachtete natürliche Sprache enthält deshalb keine Orientierungspunkte oder Wegmarken. Die Anweisungen werden mit Hilfe der Optionen aus der Karte interpretiert, um so den Suchraum einzugrenzen. Dabei wird eine modifizierte Version von [WM06] verwendet. Die Anweisungen

werden anhand von Schlüsselwörtern aufgeteilt. Das System lernt ausschließlich aus Beispielen, die angeben, wie Wegbeschreibungen in Pfade umgesetzt werden.

Chen und Mooney [CM11] stellen ein System vor, dessen Trainingsdaten nicht auf natürlicher Sprache und den entsprechenden formalen Äquivalenzen besteht, sondern aus Paaren von natürlicher Sprache und Aktionen. Die Aktionen beziehen sich hier nur auf Richtungswechsel. Orientierungspunkte oder Wegmarken werden nicht verwendet. Es wird ein *Semantic Parser* KRISP von [KM06] verwendet, der die Ableitung von unbekanntem Input sowie von mit Fehlern behaftetem Input erlaubt.

Matuszek et al. [MHZF12] erweitern [MFK10] und [CM11] so, dass die Sprachverarbeitung auch in unbekanntem Umgebungen verwendet werden kann. Außerdem werden komplexe Anweisungen wie „die nte Abzweigung“ berücksichtigt. Das System basiert auf einer kombinatorischen Kategorial-Grammatik (s. Kapitel 2.6). Für einfache Routenbeschreibungen erreicht diese Arbeit eine Interpretationsquote von ca. 66% und entspricht damit nach Matuszek der Erfolgsquote der anderen Arbeiten. Bei komplexen Anweisungen wird eine Quote von ca. 49% erreicht.

Kollar et al. [KTRN10] verwenden natürliche Sprache, um in Notfall-Szenarien Richtungsbeschreibungen zu verstehen. Dazu wird die aufgenommene Sprache nach *spatial description clauses* (SDC) geparsed und so der beschriebene Weg anhand einer Karte vollzogen. SDCs sind Baumstrukturen, die Wegbeschreibungen aufnehmen können. Jeder SDC besteht aus einem Subjekt (figure), einem Verb (verb), einem Orientierungspunkt (landmark) und einer räumlichen Relation (spatial relation). Als Trainingsbasis wurden von 15 Personen Wegbeschreibungen eingefordert. Die Wegbeschreibungen sind sequentiell, enthalten Orientierungspunkte und Beschreibungen, wie der Weg relativ zu diesen Orientierungspunkten fortzuführen ist. Die Zuordnung der einzelnen Wörter zu dieser Einteilung erfolgt durch einen Conditional-Random-Field-Parser [SP03].

Tellex et al. [TKD⁺11] erweitern das System von Kollar et al. [KTRN10]. Am Beispiel eines autonomen Gabelstaplers (s. Abbildung 2.7) verwenden sie SMT zur Verarbeitung natürlichsprachlicher Anweisungen. Die Maschine wird anhand von Trainingsbeispielen in die Lage versetzt auch bisher unbekannte Sätze zu interpretieren [Lop08]. Die natürliche Sprache wird dabei in eine formale Sprache übersetzt, die dann in einen Pfad für den Gabelstapler überführt wird. Das System versteht Anweisungen wie *Go to the first crate on the left and pick it up*. Abbildung 2.7 zeigt den in der Arbeit verwendeten Gabelstapler.

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK



Abbildung 2.7: Sprachgeführte Systeme am Beispiel eines autonomen Gabelstaplers von [TKD⁺11]. Das System kann bestimmte Stapler-Aufgaben automatisch durchführen.

Im Rahmen des „Joint-Action-Service and Technology“-Projekts (JAST) wurde unter anderem an der Technischen Universität München an der Interaktion zwischen Mensch und Roboter bei der Durchführung einer gemeinsamen Aufgabe geforscht [FGI⁺09]. Giuliani [Giu11] beschreibt im Rahmen dieses Projekts die multimodale Fusion, indem er die natürliche Sprache mit Zeige-Gesten verbindet. Die Arbeit verwendet die kombinatorische Kategorial-Grammatik und erfasst Gesten über eine fest montierte Kamera, die über der Arbeitsfläche montiert ist. Die Kombination dieser zwei Modalitäten erlaubt es Sätze, wie „Nehme diesen Würfel und diese Schraube“ richtig zu interpretieren. Stückler und Behnke [SB09] stellen ein vergleichbares Projekt *Dynamaid* mit der Ausrichtung auf multimodale Fusion vor. Das Projekt adressiert die „intuitive multimodale Kommunikation“ in der häuslichen Service-Robotik. *Dynamaid* ist eine Mechanik auf einer omnidirektionalen Basis, die z.B. in der Lage ist, Getränke zu servieren. Im häuslichen Umfeld sind die Sprach-Interaktion sowie die Verwendung von Gesten üblich. Zur Sprachverarbeitung verwendet *Dynamaid* das kommerzielle System Loquendo¹.

Wang et al. [WLY11] stellt eine mobile humanoide Roboterplattform vor, die natürlichsprachliche Kommandos in der Teleoperation entgegen nimmt. Das System verwendet die Microsoft SDK und verarbeitet die Kommandos *Begin*, *Turn left/right*, *Go Forward/Backward*, *Shake hands* und *Search*. Bei *Shake hands* hebt das System den Arm und *Search* veranlasst das System ein bestimmtes Objekt in der Umgebung zu suchen.

Das System von Stenmark und Nugues [SN13] erlaubt es Industrieroboter mit Hilfe von natürlichsprachlichen Anweisungen zu programmieren. Die natürliche Sprache wird durch einen generischen, semantischen Parser in Steuerkommandos für den Roboter überführt. Der Parser basiert auf *Penn Treebank* [MMS93, MKM⁺94] und verwendet das *Propbank* Lexikon [JN08],

¹<http://www.nuance.com/for-business/by-solution/customer-service-solutions/solutions-services/inbound-solutions/loquendo-small-business-bundle/text-to-speech/index.htm>

das für jedes (englische) Verb die typische Anzahl an Argumenten angibt. Der Parser kann anhand der gefundenen Verben die entsprechenden Argumente in Anweisungen für den Roboter zusammensetzen. Das System berücksichtigt die Verben *Take*, *Insert*, *Put* und *Calibrate*.

Die vorgestellten Systeme beziehen sich auf Wegbeschreibungen, um ein bestimmtes Ziel innerhalb eines Gebäudes zu erreichen. Für die Führung einer Kamera bedarf es anderer Kommandos, die in diesen Arbeiten nicht berücksichtigt werden.

2.6 Kombinatorische Kategorial-Grammatik (CCG)

Die vorhergehenden Abschnitte zeigen bereits, dass es unterschiedliche Möglichkeiten gibt, um natürliche Sprache zu verarbeiten. Ähnlich wie im Bereich der „reinen künstlichen Intelligenz“¹ kann die Verarbeitung der natürlichen Sprache ebenfalls symbolisch sowie neuronal geschehen. Oft bleibt es der persönlichen Überzeugung überlassen für welche Methode man sich entscheidet. In dieser Arbeit wird eine Grammatik – die kombinatorische Kategorial-Grammatik – verwendet.

Eine Grammatik beschreibt die Struktur einer Sprache im Rahmen eines symbolischen Ansatzes. Sie kann deterministisch einen natürlichsprachlichen Satz interpretieren oder nicht. Für die Verwendung in einer Studioumgebung ist deshalb eine Grammatik besser geeignet, als z.B. ein neuronaler Ansatz, weil versehentliche Interpretationen eines gesprochenen Satzes ausgeschlossen sind. Eine Grammatik kann nur zuvor definierte Kommandos ableiten.

Die Vorteile einer Grammatik liegen in der leichten Erweiterbarkeit. So können sowohl einzelne Begriffe, wie auch ganze Sätze, unter Berücksichtigung der bereits bestehenden Struktur, ergänzt werden.

Es gibt Grammatiken, die Wörter bestimmten Kategorien zuordnen und explizit Regeln formulieren, wie diese Kategorien zu gültigen Sätzen kombiniert werden dürfen (vgl. *regelbasierte Grammatiken*). Die Kategorial-Grammatik (CG) als Basis der CCG wird dagegen als *lexikalisierte Grammatik* bezeichnet. Bei diesen Grammatiken werden die Ableitungsregeln implizit durch die Kategorien mitformuliert. Die CCG erweitert die CG um die Möglichkeit, die Semantik eines Satzes in Form von kategorischer Logik (categorial logic) im Rahmen der Ableitung zu extrahieren.

Im Folgenden wird kurz die Historie der CCG umrissen und dann die kombinatorische Kategorial-Grammatik selbst eingeführt. Dabei wird erst die syntaktische Komponente der

¹Die Verarbeitung der natürlichen Sprache zählt selbst auch zum Fachbereich der künstlichen Intelligenz

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

Grammatik isoliert vorgestellt. Danach wird diese Information durch semantische Informationen ergänzt. Das Kapitel schließt mit einer kurzen Vorstellung der Java-Bibliothek OpenCCG, die in dieser Arbeit verwendet wurde. Die folgenden Abschnitte stützen sich auf die Arbeiten von v. Heusinger [Heu91], Müller [Mül10], Steedman [Ste05] und Baldrige [BK03] sowie von Steedman und Baldrige [SB11] und Giuliani [Giu06].

2.6.1 Historie

Der Ursprung der kombinatorischen Kategorial-Grammatik (engl. *combinatory categorial grammar* (CCG)) ist die klassische Kategorial-Grammatik (CG). Sie wird Ajdukewicz [Ajd67] und Bar-Hillel [Yeh53] zugeschrieben. Ajdukewicz stützt sich auf die Arbeiten von Husserl [Hus28] und von Frege [Fre18]. Husserl entwickelte die Idee der Bedeutungskategorie, in der erstmalig Worten, die untereinander ausgetauscht werden können ohne den Sinn des Satzes zu ändern, die gleiche Kategorie zugeteilt wurde. Frege betrachtete einen verbalen Satz wie eine Gleichung. In einer Gleichung gibt es bestimmte Ausdrücke, die in sich gesättigt sind und andere, die eine Erweiterung durch andere Ausdrücke benötigen. Später wird diese Grundform der Kategorial-Grammatik in die heute verwendete Form erweitert [Heu91].

2.6.2 Aufbau und Funktionsweise der Kategorial-Grammatik

Die CG basiert auf einem Lexikon, das alle verwendbaren Wörter der Sprache enthält. Die Grammatik ordnet jedem Wort der Sprache eine oder mehrere Kategorien zu. Diese können entweder *atomar* oder *komplex* sein. Atomare Kategorien werden auch als Grundkategorien bezeichnet. Zu diesen werden z.B. Nomen und allgemein alle Wörter gezählt, die semantisch relevante Information tragen. Grundkategorien stehen für sich und benötigen keine Ergänzung durch andere Wörter bzw. Kategorien.

Komplexe Kategorien beschreiben Funktionen von atomaren oder komplexen Kategorien. Dabei können sowohl das Ergebnis wie auch die Argumente selbst wieder Funktionen sein. In der vorliegenden Arbeit wird als Konvention die Rechts-Assoziation angenommen. Damit steht das Ergebnis der Funktion immer auf der linken Seite des Ausdrucks. Die Argumente schließen sich an das Ergebnis an. Die Funktion kann fordern, dass die Argumente strukturell vor bzw. hinter der aktuellen Kategorie vorhanden sein müssen. In der Grammatik wird dies durch die Richtung des Schrägstriches (engl. *Slash*) kenntlich gemacht. Der *Forwardslash* (/) bedeutet, dass hinter dem Funktor das entsprechende Argument stehen muss, während bei Verwendung eines *Backslashes* (\) das Argument des Funktors vor dem Funktor selbst erwartet

wird. Die einzelnen Ableitungsregeln werden in Kapitel 2.6.2.2 näher erläutert. Im Folgenden werden jedoch erst die Slash-Modalitäten vorgestellt. Diese sind nötig, um die anschließenden Ableitungsregeln vollständig einführen zu können.

2.6.2.1 Slash-Modalitäten

Mit Hilfe von Slash-Modalitäten kann die Verwendung der unterschiedlichen Ableitungsregeln (vgl. Kapitel 2.6.2.2) für bestimmte Kategorien eingeschränkt und so die Ableitung von grammatikalisch falschen Sätzen vermieden werden. Die Slash-Modalitäten werden jeweils durch ein bestimmtes Symbol, das an die Slash-Operatoren angehängt wird, repräsentiert. Man unterscheidet die Modalitäten \star , \times , \diamond und \bullet . So wird aus dem Nachfolge-Operator ($/$) z.B. die Form $/\star$ oder aus dem Vorgänger-Operator (\backslash) beispielsweise ein Ausdruck $\backslash\circ$. Die indizierten Symbole unterscheiden die jeweilige Anwendbarkeit der Ableitungsregeln.

- \star : Eingeschränktester Modus. Kategorien, die diesen Modus verwenden, können nur durch die Vorwärts- bzw. Rückwärts-Applikation abgeleitet werden.
- \times : Erlaubt eingeschränkte Permutation der Kategorien¹.
- \diamond : Erlaubt nur reihenfolgeerhaltende Regeln.
- \bullet : Erlaubt die Anwendung aller Regeln.

Wird keine Modalität angegeben (z.B. $/$) wird hier die relaxierteste Modalität ($/\bullet$) angenommen. Der Grundgedanke hinter den Modalitäten ist, dass jede der Ableitungsregeln von diesen Modalitäten abhängt und somit nur für diejenigen Kategorien wirken kann, die den entsprechenden Slash-Typ verwenden [BK03].

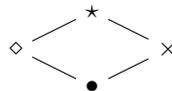


Abbildung 2.8: Hierarchie der Slash Modalitäten für Ableitungsregeln.

Abbildung 2.8 zeigt die Hierarchie der Modalitäten. \star ist die Modalität mit der stärksten Einschränkung, \bullet ist diejenige mit der geringsten Einschränkung. Sie erbt die Eigenschaften der anderen Modalitäten.

¹Ableitungsregeln für Permutation von Kategorien werden in dieser Arbeit nicht vorgestellt. Siehe dafür [SB11]

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

2.6.2.2 Ableitungsregeln

Innerhalb der komplexen Kategorien wird oft die Kategorie S verwendet. Diese steht für einen abgeleiteten Satz. Sätze sind genau dann ableitbar, wenn durch die Anwendung von Ableitungsregeln (Funktionsapplikation) die im Satz vorkommenden Kategorien auf die Kategorie S reduziert werden können. Einige Ableitungsregeln werden im Folgenden vorgestellt. Die grundlegendste Ableitungsregel ist die *Vorwärts-* bzw. *Rückwärts-Applikation*.

$$X/\star Y \quad Y \Rightarrow X \qquad (>) \qquad (2.1)$$

$$Y \quad X \backslash \star Y \Rightarrow X \qquad (<) \qquad (2.2)$$

In (2.1) wird eine komplexe Kategorie $X/\star Y$ mit Hilfe einer nachfolgenden, atomaren Kategorie Y auf die atomare Kategorie X abgeleitet. Der Ausdruck $X/\star Y$ sagt aus, dass die Kategorie eine nachstehende Kategorie Y braucht, um auf X abgeleitet werden zu können. Das Symbol \star legt weiter fest, dass diese Regel für jede Kategorie mit allen Slash-Modalitäten verwendet werden kann. Die Anwendung dieser Regel wird als *Vorwärts-Applikation* bezeichnet und in einer Ableitung durch das Symbol $(>)$ visualisiert.

In (2.2) wird eine komplexe Kategorie $X \backslash \star Y$ durch eine vorausgehende, atomare Kategorie Y zu X abgeleitet. Die Kategorie $X \backslash \star Y$ braucht also eine vorausgehende Kategorie Y um nach X abgeleitet werden zu können. Die Anwendung dieser Regel wird als *Rückwärts-Applikation* bezeichnet und in einer Ableitung durch das Symbol $(<)$ dargestellt. Kategorial-Grammatiken mit den oben genannten Funktionsapplikationen entsprechen in akzeptierender Anwendung kontextfreien Sprachen.

Mit Hilfe der Vorwärts- und Rückwärts-Applikation können nicht alle gültigen Sätze einer Sprache abgeleitet werden. An dieser Stelle werden deshalb die harmonische Komposition und die Anhebung (Typeraising) als weitere grundlegende Regeln vorgestellt. Auf die Vorstellung aller Regeln wird aus Platzgründen verzichtet. Insbesondere werden hier keine permutativen Ableitungsregeln vorgestellt.

$$X/\diamond Y \quad Y/\diamond Z \Rightarrow X/\diamond Z \qquad (>_B) \qquad (2.3)$$

$$Y \backslash \diamond Z \quad X \backslash \diamond Y \Rightarrow X \backslash \diamond Z \qquad (<_B) \qquad (2.4)$$

In (2.3) wird die harmonische Vorwärts-Komposition gezeigt. Es werden zwei komplexe Kategorien X/Y und Y/Z auf die komplexe Kategorie X/Z abgeleitet. In einer Ableitung wird

2.6 Kombinatorische Kategorial-Grammatik (CCG)

dies durch das Symbol ($>_B$) kenntlich gemacht. Die harmonische Rückwärts-Komposition ($<_B$) ist in (2.4) dargestellt. (2.5) bis (2.8) zeigen explizit die Slash-Modalitäts-Kombinationen, die aufgrund der in der Ableitungsregel verwendeten Slash-Modalität (\diamond) möglich sind.

$$X/\diamond Y \quad Y/\diamond Z \Rightarrow X/\diamond Z \quad (>_B) \quad (2.5)$$

$$X/\diamond Y \quad Y/\bullet Z \Rightarrow X/\bullet Z \quad (>_B) \quad (2.6)$$

$$X/\bullet Y \quad Y/\diamond Z \Rightarrow X/\diamond Z \quad (>_B) \quad (2.7)$$

$$X/\bullet Y \quad Y/\bullet Z \Rightarrow X/\bullet Z \quad (>_B) \quad (2.8)$$

Die harmonische Komposition kann nur durchgeführt werden, wenn die Modalitäten in den Kategorien jeweils vom Typ \diamond oder \bullet (vgl. Abbildung 2.8) und in der angegebenen Reihenfolge gegeben sind. (2.9) und (2.10) zeigen jeweils das Vorwärts-Typeraising sowie das Rückwärts-Typeraising.

$$X \quad \Rightarrow K/i(K \setminus_i X) \quad (>_T) \quad (2.9)$$

$$X \quad \Rightarrow K \setminus_i(K/i X) \quad (<_T) \quad (2.10)$$

Dabei wird eine atomare Kategorie auf eine komplexe Kategorie angehoben. Die Variable K steht für beliebige Kategorien. Der Index i zeigt an, dass für beide Slash-Operatoren die gleiche Modalität verwendet werden muss. Diese ist frei wählbar. In einer Ableitung wird Typeraising durch die Symbole ($>_T$) bzw. ($<_T$) kenntlich gemacht.

Die hier vorgestellten Ableitungsregeln bilden die Basis für die CG. Sie ermöglichen Assoziativität in den Ableitungen und sind reihenfolgeerhaltend. Für die Ableitung von sehr komplizierten Sätzen können weitere Ableitungsregeln notwendig sein. Für weiterführende Informationen siehe [SB11].

$$\begin{aligned} \textit{Katze} &\vdash N \\ \textit{Couch} &\vdash N \\ \textit{mag} &\vdash S \setminus \textit{NP}/\textit{NP} \\ \textit{die} &\vdash \textit{NP}/N \end{aligned} \quad (2.11)$$

Das Beispiel in (2.11) zeigt eine einfache Kategorial-Grammatik, die den natürlichsprachlichen Satz *Die Katze mag die Couch* ableiten kann. Dabei wurden den Nomen *Katze* und *Couch* jeweils die atomare Kategorie N zugewiesen. Dem Verb *mag* (mögen) sowie dem Artikel *die*

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

werden die komplexen Kategorien $S \setminus NP/NP$ respektive NP/N zugeteilt. Die Zuweisung der Wörter zu den entsprechenden Kategorien ist durch das Symbol \vdash repräsentiert. Abbildung 2.9 zeigt die Ableitung für obigen Beispielsatz mit Hilfe der CG. In dem Beispiel wird dreimal die Vorwärts-Applikation und einmal die Rückwärts-Applikation angewendet.

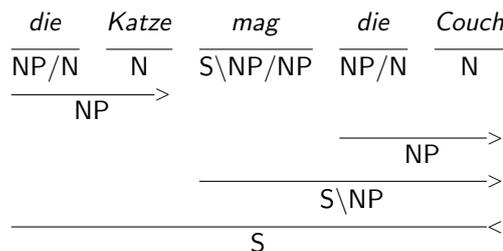


Abbildung 2.9: Ableitung von *Die Katze mag die Couch* mit Hilfe der CCG. Abgeleitete Konstituenten (Satzfragmente) sind unterstrichen dargestellt. Die resultierende Kategorie steht unterhalb des Strichs. Das Symbol am Ende des Strichs gibt die verwendete Ableitungsregel an.

Abgeleitete Konstituenten (Satzfragmente) sind in der Darstellung unterstrichen visualisiert. Die resultierende Kategorie steht unterhalb des Strichs. Das Symbol am Ende des Strichs gibt die verwendete Ableitungsregel an.

2.6.3 Semantik in der CCG

Bei Verwendung einer CCG kann die Semantik eines Satzes ausgehend von der Syntax bestimmt werden. Aus diesem Grund eignet sich die CCG besonders zur Verarbeitung natürlicher Sprache. Die Semantik wird dann durch eine kombinatorische Logik ausgedrückt. Einem Wort wird dafür neben der Ableitungsvorschrift der Kategorie auch die semantische Interpretationsvorschrift mitgegeben. Diese wird im Laufe der Ableitung der Konstituenten mit semantischer Information gefüllt. Die Ableitungsregeln (2.1) bis (2.4) sowie (2.9) und (2.10) werden wie folgt ergänzt:

$$X /_{\star} Y : f \quad Y : a \Rightarrow X : fa \quad (>) \quad (2.12)$$

$$Y : a \quad X \setminus_{\star} Y : f \Rightarrow X : fa \quad (<) \quad (2.13)$$

$$X /_{\diamond} Y : f \quad Y /_{\diamond} Z : g \Rightarrow X /_{\diamond} Z : \lambda x.f(gx) \quad (>_B) \quad (2.14)$$

$$X \setminus_{\diamond} Y : g \quad Y \setminus_{\diamond} Z : f \Rightarrow X \setminus_{\diamond} Z : \lambda x.f(gx) \quad (<_B) \quad (2.15)$$

$$X : a \Rightarrow T /_i(T \setminus_i X) : \lambda f.f a \quad (>_T) \quad (2.16)$$

$$X : a \Rightarrow T \setminus_i(T /_i X) : \lambda f.f a \quad (<_T) \quad (2.17)$$

2.6 Kombinatorische Kategorial-Grammatik (CCG)

In Beispiel (2.18) wird, angelehnt an [Ste05], die semantische Vorschrift mit Hilfe der λ -Schreibweise anhand des Verbs *mag* vorgestellt. Diese sagt aus, dass *mag* eine Relation von zwei Objekten darstellt, für die gilt $mag'(x, y)$. Die λ -Notation erlaubt dabei den „Zugriff auf offene Stellen in komplexen semantischen Repräsentationen“ [Mül10]. In CCG kann so durch die logische Ergänzung der Nominalphrasen *NP* x und y ersetzt werden. Es entsteht ein logischer Ausdruck, der die semantische Aussage des Satzes wiedergibt.

$$\begin{aligned}
 & Katze \vdash N : Obj'(Katze) \\
 & Couch \vdash N : Obj'(Couch) \tag{2.18} \\
 & die \vdash NP/N : \lambda x. defArt'(x) \\
 & mag \vdash S \backslash NP/NP : \lambda y. \lambda x. mag'(x, y)
 \end{aligned}$$

Alle logischen Bezeichner sind in (2.18) mit einem Apostroph (') gekennzeichnet. So können sie klar von den semantischen Variablen unterschieden werden. Die Nomen *Katze* und *Couch* werden jeweils als Objekte *Obj'* markiert. Der Artikel *die* gibt an, dass ein bestimmtes Objekt gemeint ist. Dies ist durch den logischen Bezeichner *defArt'* ausgedrückt. Für die gegebenen Vorschriften ergibt sich für *die Katze* bzw. *die Couch* die logische Form: $defArt'(Obj'(Katze))$ bzw. $defArt'(Obj'(Couch))$.

$$\begin{array}{c}
 \begin{array}{ccc}
 \frac{mag}{S \backslash NP/NP : \lambda y. \lambda x. mag'(x, y)} & \frac{die}{NP/N : \lambda x. defArt'(x)} & \frac{Couch}{N : Obj'(Couch)} \\
 & & \xrightarrow{\hspace{1.5cm}} \\
 & & NP : defArt'(Obj'(Couch)) \\
 \xrightarrow{\hspace{1.5cm}} & & \\
 S \backslash NP : \lambda x. mag'(x, defArt'(Obj'(Couch))) & &
 \end{array}
 \end{array}$$

Abbildung 2.10: Syntaktische und semantische Ableitung von *mag die Couch* anhand der CCG. Der fehlende Konstituent *Die Katze* wurde aus Platzgründen weggelassen. Der vollständig logische Ausdruck ist leicht weiter abzuleiten.

Abbildung 2.10 zeigt die Entwicklung der logischen Form für die Ableitung am bereits eingeführten Beispiel. Dabei fehlt die Entwicklung von *Die Katze*. Ausgehend von der bereits erreichten logischen Form kann leicht die endgültige Form der Ableitung nachvollzogen werden. Sie ergibt sich zu: $mag'(defArt'(Obj'(Katze)), defArt'(Obj'(Couch)))$.

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

2.6.4 OpenCCG

In dieser Arbeit wird die kombinatorische Kategorial-Grammatik in Form der Java-Bibliothek OpenCCG¹ verwendet. Neben der Möglichkeit natürlichsprachliche Sätze abzuleiten und in eine logische Form zu wandeln, steht die OpenCCG für eine hohe Verarbeitungsgeschwindigkeit.

Die OpenCCG basiert auf Mark Steedmans CCG-Formalismus und wurde durch Jason Baldrige, Geert-Jan Kruijff und Michael White [Whi06] weiterentwickelt. Die Bibliothek entstand aus dem Grok-System und ist in der Lage einen natürlichsprachlichen Ausdruck in eine formale Hybrid-Logik zu übersetzen. Anders als die bisher vorgestellte kombinatorische Logik auf Basis der Lambda-Notation, erlaubt die Hybrid-Logik eine leichtere Weiterverarbeitung. So kann der Hybrid-Logische Ausdruck auch in XML ausgegeben werden.

Die Konfiguration der OpenCCG geschieht über verschiedene XML-Dateien, wobei jede dieser Dateien einen Teil der Grammatik enthält. Auf die Vorstellung der einzelnen Konfigurationsdateien wird an dieser Stelle verzichtet, weil das Format der Dateien syntaktisch sehr aufwendig gestaltet ist. Mit Hilfe eines Tools, das von Jason Baldrige [BCPW07] entwickelt wurde, kann die Grammatik deutlich intuitiver gestaltet werden. Das Tool nennt sich DotCCG und findet in dieser Arbeit Anwendung. Mit Hilfe der DotCCG können Kategorien in CCG-Notation beschrieben werden.

Die folgenden Kapitel stellen zuerst die *features* und die *word*-Struktur von DotCCG vor. Daraufhin wird die Hybrid-Logik anhand der Arbeit von Baldrige und Kruijff [BK02] vorgestellt. Ein vollständiges Beispiel, das die syntaktische und semantische Ableitung des einfachen Stopp-Kommandos zeigt, schließt das Kapitel. Anhand des Beispiels wird ebenfalls die XML-Struktur vorgestellt, die von der OpenCCG zur maschinellen Weiterverarbeitung angeboten wird.

2.6.4.1 Eigenschaften von Wörtern in OpenCCG - *features*

Features können in der OpenCCG eine syntaktische oder eine semantische Funktion besitzen. Im syntaktischen Fall helfen die Features nur syntaktisch korrekte Satzgefüge abzuleiten. Im semantischen Fall können mit Hilfe von Features den einzelnen Wörtern bestimmte semantische Eigenschaften mitgegeben werden, die später in der logischen Formel verfügbar sind. Im Folgenden werden die Abschnitte **features** und **words** der DotCCG vorgestellt. Sie repräsentieren die wichtigsten Abschnitte der DotCCG Datei. Für die weiteren Möglichkeiten innerhalb von DotCCG siehe [BCPW07].

¹<http://openccg.sourceforge.net>

```
feature{
  case<1,6,7,15> : caseInf caseNom caseGen caseDat caseAkk;
  num<2,5,6,7,15> : sg pl;
  gender<3,5,6,15>: genderFem genderMasc genderNeut;
  pers<4,15> : 1st 2nd 3rd;
  komp<8,15> : positiv, comparativ, superlativ;
  best<9,15> : def, undef;
}
```

Abbildung 2.11: Typische Merkmale für die deutsche Grammatik. Neben dem Fall (**case**) werden die Eigenschaften Numerus (**num**), Geschlecht (**gender**), die Deklination (**pers**), die Komparationsform (**komp**) sowie für Artikel festgehalten ob es ein bestimmter oder unbestimmter Artikel ist (**best**). Die Zahlen in spitzen Klammern geben an welchen Gruppen (feature-id) die jeweilige Eigenschaft zugeordnet ist.

In der DotCCG-Datei werden Merkmale von Wörtern innerhalb des **feature**-Blocks definiert. Dort werden sowohl syntaktische als auch semantische Features erfasst. Abbildung 2.11 zeigt beispielhaft Features, die für die deutsche Grammatik von Bedeutung sind. Aufgezählt sind der Kasus (**case**), der Numerus (**num**), das Geschlecht, die Person(**pers**) sowie die Komparationsform (**komp**) und die Bestimmtheit (**best**) als Eigenschaft für Artikel. Die Zahlen in den spitzen Klammern definieren unterschiedliche Gruppen. **case** ist in den Gruppen 1, 7, 6 und in Gruppe 15 enthalten. **num** ist in den Gruppe 2 und 5 enthalten etc. Für die Ableitungsregeln kann so festgelegt werden, welche Feature-Gruppen überprüft werden sollen, um unsinnige Ableitungen zu vermeiden.

2.6.4.2 Zuweisung von Kategorien zu Wörtern - *words*

Wörter werden in OpenCCG in Familien klassifiziert. Innerhalb der Familien können unterschiedliche Kategorien definiert werden. Für die jeweiligen Wörter sind alle Kategorien der Familie verwendbar. Dem Schlüsselwort **word** folgt das in der Sprache verwendbare Wort. Ihm folgt der Bezeichner für die Familie des Wortes (hier: **noun**) sowie die Aufzählung der für das Wort gültigen Features (**sg** und **genderFem**). Abbildung 2.12 zeigt beispielhaft die Angaben für das Wort *Katze*.

Katze wird der Familie **noun** zugeordnet. Dabei erhält es die Features **sg** (Singular) und **genderFem** (Feminin). Die Familie **noun** enthält eine Kategorie-Definition. Sie wird eingeleitet durch das Schlüsselwort **entry**, gefolgt von der Ableitungsvorschrift des Wortes. In spitzen Klammern wird die Eigenschaftsgruppe definiert. Gruppe 15 besagt, dass alle verfügbaren Ei-

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

```
word Katze : noun : sg genderFem;
family noun {
  entry: N<15> [X] : X: Obj (*);
}
```

Abbildung 2.12: Die Definition des Wortes `Katze` in OpenCCG. `Katze` wird der Familie `noun` zugeordnet und die Features `sg` (Singular) und `genderFem` (weiblich) zugewiesen. Die Familie wird durch `family` eingeleitet und erlaubt es unterschiedliche Ableitungsmöglichkeiten für die dazugehörigen Wörter zu definieren. Hier wird für alle Wörter der Familie `noun` nur die Ableitung auf die Kategorie `N` sowie die semantische Ableitung `X: Obj(*)` angegeben.

genschaften überprüft werden sollen. In eckigen Klammern steht der logische Bezeichner für die Kategorie. Dieser wird in der nachstehenden Vorschrift (`X: Obj(*)`) wieder aufgenommen und zusätzlich der Bezeichner `Obj` eingeführt. Der Stern sagt aus, dass das Wort selbst in die logische Form eingesetzt werden soll.

2.6.4.3 Hybrid-Logik zum Ausdruck semantischer Information

Die modale Logik war lange Zeit das Standard-Verfahren um die semantische Aussage von Sätzen mittels CCG in eine logische Form abzuleiten. Dieser Ansatz ist jedoch in Bezug auf sprachliche Belange und in Bezug auf Berechenbarkeit nicht optimal [CFPS05, Kru01]. So ist es durch modale Logik nicht möglich einen bestimmten Zeitpunkt für die Gültigkeit einer Aussage auszudrücken. Dies macht diese Form der Logik nicht passend für „praktische Anwendungen und Wissensrepräsentation“ [BK02]. Hybrid-Logik erweitert die modale Logik und erhält dabei die Entscheidbarkeit sowie die Komplexität. Hybrid-Logik führt sogenannte *Nominale* ein. *Nominale* sind einfache Ausdrücke, die es ermöglichen Zustände in der Sprache zu beschreiben. Die Ausdrücke der Hybrid-Logik können sowohl bool'sche Operatoren sowie den *Satisfaction-Operator* (`@`) enthalten. `@` gibt den Zeitpunkt der Gültigkeit für den folgenden Ausdruck an. So sagt ein Ausdruck `@iP` aus, dass der Ausdruck `P` im Zustand `i` Gültigkeit besitzt. Mit *Nominalen* können explizit Referenzen zu einem Zustand generiert werden, bei dem der jeweilige Ausdruck Gültigkeit besitzt. Die einzelnen Attribute eines Ausdrucks werden durch Konjunktion (logische Und-Verknüpfung) aneinander gereiht. Hybrid-Logik ermöglicht es zwei essentielle Bedeutungsaspekte in klarer und kompakter Weise auszudrücken: ontologische Vielfalt und die Möglichkeit der Referenzierung [BK02].

$$\begin{aligned} \text{Kamera1 Stopp} \Rightarrow @_{action}(\text{stop} \wedge \\ \langle \text{INTERJECTION} \rangle (k_2 \wedge \text{Stopp}) \wedge \\ \langle \text{ACTOR} \rangle (k_1 \wedge \text{Kamera1})); \end{aligned} \quad (2.19)$$

(2.19) zeigt, wie das einfache Kommando *Kamera1 Stopp* in Form der Hybrid-Logik ausgedrückt wird. Der logische Ausdruck ist eine Aktionsaussage, die durch $@_{action}$ eingeleitet wird. Der Typ der Aktion ist $stop^1$ und wird an erster Stelle im Ausdruck angegeben. Daraufhin werden die Attribute des Kommandos, durch Nominale markiert, in den Ausdruck aufgenommen. $\langle \text{ACTOR} \rangle$ beschreibt den angesprochenen Akteur des Kommandos. Diesem wird sowohl das jeweilige Wort aus dem Kommando *Kamera1* sowie ein Bezeichner k_1 zugewiesen. Gleiches gilt für das Nominal $\langle \text{INTERJECTION} \rangle$.

2.6.4.4 Vom Satz zum logischen Ausdruck

In den vorstehenden Kapiteln wurde das Rüstzeug vorgestellt, mit dem ein natürlichsprachlicher Satz in einen logischen Ausdruck gewandelt werden kann. Die einzelnen Schritte sollen nun anhand des einfachen Beispiel-Kommandos *Kamera1 Stopp* zusammen gestellt werden.

```

1 word Kamera1 : noun_add : sg genderFem;
2 family noun_add {
3   entry: ACTOR<15> [X] : X:actor(*);
4 }
5 word Stopp : interjection;
6 family interjection {
7   entry: S[E] \ ACTOR[X] : E:action(stop <ACTOR> X <INTERJECTION>* );
8 }

```

Abbildung 2.13: Kurzes Beispiel, wie mit der DotCCG (OpenCCG) das natürlichsprachliche Kommando *Kamera1 Stopp* in einen logischen Ausdruck umgewandelt wird.

Die zwei Wörter *Kamera1* und *Stopp* werden jeweils einer eigenen Kategorie zugewiesen. In Zeile 1 wird *Kamera1* der Kategorie `noun_add` zugewiesen. Außerdem erhält das Wort *Kamera1* die *features* `sg` und `genderFem`. Anhand dieser Features kann sichergestellt werden, dass z.B. *die Kamera1* als gültiger Ausdruck der Sprache abgeleitet werden kann, während *das Kamera1* nicht abgeleitet wird, weil das neutrale Geschlecht des Artikels nicht zum weiblichen Geschlecht

¹Die Typisierung geschieht hier in Englisch. Deshalb wird das englische *stop* (mit einem p) verwendet.

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

der *Kamera1* passt. Zeile 2 beschreibt die Kategorie `noun_add`. Jedem Wort kann eine Kategorie zugewiesen werden und jede Kategorie kann mehrere mögliche Ableitungsvorschriften enthalten. In diesem Beispiel enthält jede Wort-Familie nur eine mögliche Ableitungsvorschrift. Sie wird durch das Schlüsselwort `entry` eingeleitet und besagt, dass das Wort *Kamera1* der Kategorie `ACTOR` zugewiesen wird. Außerdem werden dieser Kategorie die Eigenschaftsgruppe 15 (vgl. Kapitel 2.6.4.1) sowie die semantische Variable `X` zugewiesen. Nach dem Doppelpunkt wird die semantische Behandlung der Kategorie definiert. In diesem Fall wird festgelegt, dass die Kategorie `ACTOR` den Bezeichner `actor` zugewiesen bekommt.

Der `*` sagt aus, dass das in dem Kommando verwendete Wort (hier: *Kamera1*) in die logische Form übernommen werden soll. In Hybrid-Logik-Notation entspricht dieser Ausdruck $@_{actor}(Kamera1)$. Zeile 7 zeigt, dass von der `interjection` die Aussage des Kommandos ausgeht. Die Kategorie des Worts `Stopp` definiert sich zu $S[E] \setminus ACTOR[X]$. Sie sagt aus, dass syntaktisch genau dann ein Satz abgeleitet werden kann, wenn der `interjection` ein Wort der Kategorie `ACTOR` vorausgeht. Semantisch bekommt die Aussage des Satzes `S` die Variable `E` und `ACTOR` die Variable `X` zugewiesen. Der Doppelpunkt leitet wieder die semantische Behandlung ein. Der Ausdruck ergibt eine Aktion vom Typ `stop`, die durch die Nominale `<INTERJECTION>` und `<ACTOR>` näher beschrieben wird. Das Ergebnis dieses Ausdrucks ist in Abbildung 2.14 dargestellt.

```
@w1:action(stop ^
    <INTERJECTION>STOP ^
    <ACTOR>(w0:actor ^ Kamera1 ^
        <num>sg-X ^
        <gender>genderFem-X))
```

Abbildung 2.14: Eine durch OpenCCG generierte logische Form für das Beispiel *Kamera1 Stopp* (vgl. Abbildung 2.13). Der *Satisfaction-Operator* (`@w1:action`) gibt an, dass es sich um eine Aktion `action` handelt. `w1` ist dabei ein automatisch generierter Bezeichner. Der Typ der Aktion ist `stop`. Außerdem wird die Aussage durch die Nominale `<INTERJECTION>` und `<ACTOR>` näher beschrieben. `<num>` und `<gender>` bezeichnen den `<ACTOR>` genauer und werden von OpenCCG automatisch durch die *feature*-Definition generiert.

OpenCCG fügt dabei automatisch eigene Bezeichner in den logischen Ausdruck ein. `w1` und `w0` sind Beispiele dafür. Außerdem werden die *feature*-Ausdrücke automatisch in die semantische Form übernommen. So erscheinen `<num>` und `<gender>` in der Ausgabe, obwohl sie vorher nicht explizit in der logischen Form definiert wurden. Für die praktische Verwendung von OpenCCG

wird eine XML-Schnittstelle angeboten. Der in Abbildung 2.14 dargestellte logische Ausdruck wird durch die in Abbildung 2.15 dargestellte XML-Struktur repräsentiert.

```
<xml>
  <lf>
    <satop nom="w1:action">
      <prop name="stop"/>
      <diamond mode="INTERJECTION">
        <prop name="STOP"/>
      </diamond>
      <diamond mode="ACTOR">
        <nom name="w0:actor"/>
        <prop name="Kamera1"/>
        <diamond mode="num">
          <prop name="sg-X"/>
        </diamond>
        <diamond mode="gender">
          <prop name="genderFem-X"/>
        </diamond>
      </diamond>
    </satop>
  </lf>
  <target>Kamera1 Stop</target>
</xml>
```

Abbildung 2.15: Durch OpenCCG generierte XML-Struktur. Die Struktur zeigt den logischen Ausdruck aus Abbildung 2.14 im XML-Format.

Der Ausdruck beginnt mit dem Tag `lf` für *logische Form* gefolgt von `satop` dem *Satisfaction-Operator*. Diesem schließt sich der Typ des Ausdrucks in Form des Elements `prop` an. Daraufhin werden die einzelnen Nominale in Form von `diamond`-Strukturen aufgezählt. Diese können auch rekursiv aufgeführt werden, wie das Beispiel des *features num* zeigt. Das Element `target` gibt noch einmal den natürlichsprachlichen Satz aus.

2. STAND DER TECHNIK UND WISSENSCHAFTLICHER ÜBERBLICK

Kapitel 3

Ausgewählte Aspekte bei Nachrichten- und Magazinsendungen

In dieser Arbeit wird die Idee von A. M. Turing verfolgt [Tur50] und Intelligenz als menschenähnliches Verhalten interpretiert. Von dieser Annahme ausgehend werden in diesem Kapitel anhand von Aspekten die Rahmenbedingungen und das Anforderungsprofil einer intelligenten Steuerung für ein Kamerasystem vorgestellt. Das Kapitel ist in theoretische Aspekte, praktische Aspekte und der Untersuchung einzelner Nachrichtensendungen aufgeteilt.

Im Rahmen der theoretischen Aspekte werden wichtige Begriffe und Schlagwörter aus dem Bereich Film- und Fernsehen erklärt. Diese Begriffe werden im weiteren Verlauf dieser Arbeit immer wieder verwendet und bilden die Grundlage zum Verständnis eines intelligent gesteuerten Kamerasystems.

Im Abschnitt zu den praktischen Aspekten werden die Kommunikationsform zwischen Regie und Kamera und der Ablauf einer realen Magazinsendung *Zur Sache Baden-Württemberg* vorgestellt. Außerdem werden hier die Ergebnisse der Feldstudie zu praktischen Anforderungen an die Kameraführung vorgestellt.

Das Kapitel schließt mit einem Vergleich und der Analyse von den einschaltquotenstärksten Nachrichtensendungen der Sender ARD, ZDF und RTL. Die Gegenüberstellung dieser Sendungen offenbart deren gestalterische Unterschiede.

Aus den hier vorgestellten Aspekten werden im nächsten Kapitel die Anforderungen einer intelligenten Kamerasteuerung abgeleitet. Der Rahmen dieser Arbeit beschränkt sich auf eine

Auswahl wichtiger Aspekte der Kameraführung.

3.1 Theoretische Aspekte der Kameraführung

In Filmproduktionen wurden die ersten Kameras eingesetzt und so die ersten Erfahrungen mit der Kameraarbeit gesammelt. Dabei haben sich Grundsätze entwickelt, die bis heute Bestand haben. Allgemein erfüllt die Kamera drei (dramaturgische) Aufgaben [Arm97]:

- Sie fotografiert das Geschehen,
- sie schafft angemessene Gewichtung und
- sie schafft Stimmung und Atmosphäre.

Das Abbilden und Festhalten bestimmter Szenen gehört unbestritten zu den grundlegenden Eigenschaften einer Kamera. Durch geschickte Kameraführung kann ferner die Gewichtung innerhalb der Szene unterstützt werden. Protagonisten werden so dargestellt, dass sie die Aufmerksamkeit des Zuschauers auf sich lenken. Letzteres geschieht durch ästhetische Gestaltungsregeln, die hier nicht weiter behandelt werden. Außerdem kann die Handhabung einer Kamera die Stimmung einer Szene beeinflussen. Für Nachrichtensendungen gelten andere Ziele als für Filmproduktionen. Allerdings werden einige der Gesetzmäßigkeiten aus dem Filmbereich auch hier angewandt. In den folgenden Kapiteln wird ein kurzer historischer Überblick zur Kameraführung gegeben. Daran schließen sich typische Einstellungsgrößen, die Wirkung der Kameraorientierung sowie typische Hilfsmittel bei der Einrichtung einer Einstellung und wesentliche Grundlangen von Kamerafahrten an.

3.1.1 Kameraführung im Wandel der Zeit

Die Möglichkeiten der Kameraführung haben sich im Laufe der Zeit mit der Entwicklung der Technologie gewandelt. Während zu Stummfilmzeiten die meist sehr großen und schweren Kameras fest auf einer Position verankert waren, änderte sich das später mit Aufkommen der deutlich leichteren und technisch besseren Kameras. Der Trend zur Miniaturisierung geht immer weiter. Mittlerweile wird mit bis zu 100 digitalen Video-Kameras (DV-Kameras) am Set gedreht [HA05].

Am Anfang war der Zuschauer ein auf eine feste Position beschränkter Beobachter des gesamten Sets. Schauspieler bewegten sich von einer Seite zur anderen durch das Bild und kamen

näher an die Kamera heran bzw. traten selten weiter in den Hintergrund. *Totale-Einstellungen* (vgl. Kapitel 3.1.2) waren die Standard-Darstellung dieser Filme. Die Kamera konnte den Zuschauer nicht durch die Handlung führen, konnte keine Emotionen bzw. Spannung aufbauen oder den Zuschauer auf bestimmte Dinge aufmerksam machen [Arm97]. Später in den 20er und 30er Jahren wurde, beginnend bei einer *Totalen-Einstellung* im Laufe der Handlung, die Kamera immer näher an das Geschehen herangerückt, bis man im Höhepunkt der Handlung eine *Nahe-Einstellung* (vgl. Kapitel 3.1.2) erreicht hatte. Die Szenen wurden dynamischer und interessanter und der Zuschauer wurde zum Teilnehmer an der sich entfaltenden Handlung [Arm97]. Das Seh-Erlebnis für den Zuschauer ist bei Sendungen mit unterschiedlichen Einstellungen größer als bei Sendungen, die nur mit einer Einstellung aufgenommen wurden. Die unterschiedlichen Eindrücke erhalten die Aufmerksamkeit des Zuschauers. Wie für den Film gilt auch für Nachrichten, dass die Sendung durch unterschiedliche Einstellungen für den Zuschauer interessanter wird.

3.1.2 Einstellungsgrößen

Im Arbeitsfeld der Kameraführung und Regiearbeit haben sich feste Definitionen für Einstellungsgrößen etabliert. Da sich diese Definitionen nicht immer decken und oft zwischen einer drei-, fünf- bzw. sieben-stufigen Einteilung unterschieden wird, wird in dieser Arbeit eine fünf-stufige Unterteilung vorgestellt, die sich hier als sinnvoll erwiesen hat. Einstellungsgrößen werden gemeinhin relativ zu Personen angegeben. Sind keine Menschen im Bild dargestellt, wird die Größe sinngemäß übertragen [Ari00].

Die *Totale-Einstellung*

Im Filmbereich gibt die *Totale-Einstellung* (oder *Totale*) dem Zuschauer einen Überblick über die Szene. Sie zeigt den Schauplatz, an dem die Handlung spielt, die Tageszeit, Wetterinformation etc. [HA05]. Die *Totale-Einstellung* dient der Einführung in neue Abschnitte eines Filmes. Sie etabliert neue Situationen und wird deshalb *Establishing Shot* genannt [Arm97]. Abbildung 3.1 (a) zeigt schematisch eine *Totale-Einstellung*.

Im Nachrichtenbereich haben *Totale-Einstellungen* ebenfalls einführenden bzw. verabschiedenden Charakter. Gerade in imposant gebauten Studios wird zur Einführung gern das gesamte Ausmaß des oft virtuellen Studios gezeigt. Der Zuschauer sieht, wie viele Moderatoren an der aktuellen Sendung teilnehmen und wo sich die Moderatoren im Studio befinden. Am Ende einer

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN



(a) *Totale-Einstellung*. Man sieht neben dem Akteur auch noch die Umgebung in der er sich befindet.



(b) *Zweier-Einstellung*. Beide Moderatoren werden gezeigt.



(c) *Amerikanische-Einstellung*. Der Akteur wird bis über das Knie gezeigt.



(d) *Halbnahe-Einstellung*. Der Akteur wird bis zum Gürtel gezeigt.



(e) *Nahe-Einstellung*. Der Akteur wird bis unter die Schultern gezeigt.

Abbildung 3.1: Schematische Darstellung der Haupteinstellungsgrößen im Nachrichtbereich.

Sendung wird nach der Verabschiedung wieder die *Totale* gezeigt. *Totale-Einstellungen* rahmen eine Nachrichtensendung ein.

Die *Zweier-Einstellung*

Der *Zweier-Einstellung* kommt im Nachrichtbereich eine besondere Bedeutung zu. Während sie im Film den *Nahen-Einstellungen* zugesprochen wird, wird sie im Rahmen des Nachrichten-Umfeldes den *Totalen-Einstellungen* zugeordnet. Diese Einstellung wird für Dialog-Situationen und speziell bei Nachrichten für Übergaben verwendet. In Dialogen werden bei dieser Ein-

stellung beide Gesprächspartner gleichzeitig gezeigt. Bei Übergaben wird diese Einstellung als „Zwischeneinstellung“ verwendet, um beide Moderatoren zu zeigen. Sie zeigt dem Zuschauer, an wen die Sendung übergeben wird und wo sich diese Person im Studio befindet. Abbildung 3.1 (b) zeigt schematisch eine sehr enge *Zweier-Einstellung*.

Es gibt zwei Möglichkeiten eine *Zweier-Einstellung* zu zeigen. Aus dem Filmbereich ist die Over-the-Shoulder Einstellung bekannt. Dabei wird ein Protagonist von hinten und der andere von vorne gezeigt. Diese Einstellung wurde lange Zeit bei Nachrichtensendung vermieden. Konventionell für Nachrichten ist die Side-By-Side Einstellung. Hier wird durch Videotechnik das aktuelle Bild reduziert und ein zweites Bild neben das eigentliche Bild gestellt. Welche der beiden Darstellungsformen für eine *Zweier-Einstellung* verwendet wird, hängt von den technischen Möglichkeiten und den Vorlieben der Redaktion bzw. des Senders / Studios ab. Eine Side-By-Side wirkt weniger natürlich, zeigt den Moderator aber nicht von hinten.

Die Amerikanische-Einstellung

Die *Amerikanische-Einstellung* zeigt Personen bis knapp über das Knie. Im übertragenen Sinn wird die jeweilige Person „mit ihrem Revolver“ gezeigt. Nach [HA05] tritt die Bedeutung der Umgebung hier zunehmend in den Hintergrund. Eine solche Einstellung wird im Nachrichtenbereich für das Wetter oder auch bei Sende-Abschnitten, in denen der Moderator sich stark bewegt (z.B. Erklärraum) verwendet. Abbildung 3.1 (c) zeigt schematisch eine solche *Amerikanische-Einstellung*.

Die Halbnahe-Einstellung

In der *Halbnahen-Einstellung* wird der Moderator an seinem Arbeitsplatz gezeigt. Man sieht im Wesentlichen den Oberkörper des Moderators. Zusätzlich kann, sofern vorhanden, ein Teil des Moderatorentischs und/oder ein Themenbild im Hintergrund eingeblendet sein. Die meisten Meldungen werden in dieser Einstellung präsentiert. Sie ermöglicht es sowohl Emotion als auch Information zu übertragen. Abbildung 3.1 (d) zeigt eine solche *Halbnahe-Einstellung*. Auf ein Hintergrundbild oder einen Moderatorentisch wurde in diesem Beispiel verzichtet.

Die Nahe-Einstellung

Die *Nahe-Einstellung* zeigt einen Menschen vom Kopf bis unter die Schultern. Laut [HA05] hat diese „Einstellungsgröße [...] subjektive und emotionale Qualitäten. Sie bringt dem Zuschauer

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

die jeweiligen Personen nahe und vermag so Identifikation zu simulieren“. In Nachrichtensendungen werden besonders emotionale Meldungen bevorzugt in dieser Einstellung gezeigt. Die *Nahe* kann auch zur Auflockerung einer sonst sehr statischen, ausschließlich in der *Halbnahen* präsentierten Sendung verwendet werden.

Nahe-Einstellungen werden immer dann gern verwendet, wenn sich der Akteur an den Zuschauer wendet und in die Kamera schaut. Üblich ist dies bei Werbespots, in Lehr- und Demonstrationsfilmen, in Nachrichtensendungen und Gameshows [HA05]. Abbildung 3.1 (e) zeigt schematisch eine Nahe-Einstellung.

Das hier vorgestellte fünf-stufige Modell ist nicht bindend. Sowohl die Unterteilung der einzelnen Standard-Einstellung als auch das Verständnis, was unter einer bestimmten Einstellung zu verstehen ist, kann sich von Sender zu Sender unterscheiden.

3.1.3 Kamerablickwinkel

Die Kamera soll das Geschehen aus den Augen des Betrachters zeigen. Dies ist im Normalfall horizontal auf Augenhöhe eines Erwachsenen. Die horizontale Kamera generiert einen neutralen Eindruck. Der Zuschauer ist es gewohnt seine Umwelt in dieser Perspektive wahrzunehmen [HA05]. Um besondere Wirkungen beim Betrachter zu provozieren, kann von dieser neutralen Position abgewichen werden. Die Kamera wird dann um die Längsachse gekippt (Roll) oder unter bzw. über die Augenhöhe des Protagonisten gestellt.

Seitlich geneigte Kamera

Eine um die Längsachse gekippte Kamera suggeriert dem Zuschauer Ungleichgewicht - etwas stimmt in einer solchen Szene nicht. Diese Stilform ist auch im Filmbereich sehr ungewöhnlich und selten. Sie unterstellt, dass die Welt in eine Schiefelage geraten ist und keine Normalität vorliegt [Arm97]. In manchen Fällen wird eine seitlich gekippte Kamera auch für Spezialeffekte eingesetzt. So ist es zum Beispiel leichter eine Kamera zu kippen als das ganze Set. Bei Aufnahmen im Inneren eines Schiffes, eines U-Bootes oder während eines Erdbebens wird dieses Stilmittel gern eingesetzt. Die Wankbewegungen der Szene sind mit Hilfe der Kamera deutlich leichter umzusetzen als mit dem Set selbst [Ari00]. Abbildung 3.2 (a) zeigt eine Aufnahme mit seitlich geneigter Kamera.

Im Nachrichten- und Magazinbereich werden seitlich geneigte Kameras nicht verwendet.



(a) Seitlich gekippte Kamera wirken wie eine verdrehte Welt.



(b) Blick von oben auf einen Akteur. Der Akteur wirkt dadurch unterwürfig.



(c) Blick von unten auf einen Akteur. Der Akteur wirkt dadurch dominant.

Abbildung 3.2: Unterschiedliche Wirkungen von Einstellungen. Aufnahme (a) zeigt eine Einstellung mit seitlich geneigter Kamera (Roll), (b) zeigt eine aufsichtig aufgenommene Einstellung und (c) eine Einstellung mit Untersicht.

Aufsichtige und untersichtige Aufnahmen

Ein Effekt der aufsichtigen Darstellung ist, dass sie den Wert der angeschauten Person verringert. Die jeweilige Person wirkt unterwürfig, einsam oder armselig [Arm97, HA05, Kan03]. Abbildung 3.2 (b) zeigt schematische eine aufsichtig aufgenommene Person.

Untersichtige Aufnahmen ordnen den Betrachter dem Gesehenen unter. Personen, die untersichtig aufgenommen werden, wirken deshalb „selbstbewusst, heldenhaft oder überlegen“ [Kan03, HA05, Kuc05]. Abbildung 3.2 (c) zeigt schematisch eine solche Einstellung.

In Nachrichtensendungen wird, soweit möglich, die neutrale Ausrichtung der Kamera verwendet. Das Ziel von Nachrichten ist es neutral zu informieren. Eine untersichtige oder aufsichtige Einstellung würde diesem Ziel entgegen stehen.

3.1.4 Safe-Title und Safe-Action

Safe-Title und *Safe-Action* begrenzen die Darstellungsfläche im aufgenommenen Videobild. Sie werden durch projizierte Rechtecke im intern verwendeten Videobild dargestellt. Der Zuschauer kann diese Rechtecke nicht sehen. Diese technische Vorgabe gewährleistet, dass Geräte, die nicht genau der Norm in Höhe und Breite entsprechen, trotzdem das Wesentliche des Bildes zeigen können, ohne Teile des Bildes abzuschneiden. Sie bieten eine Orientierung für die Platzierung des Moderators oder von Logos, Bauchbinden¹ etc. Die Empfehlung gibt das IRT² für Deutschland heraus. Für die öffentlich-rechtlichen Sendeanstalten liegt bei HD-Produktionen der Safe-Title-Bereich innerhalb von je 5% vertikalen und je 10% horizontalen Seitenrändern [Ins10]. Abbildung 3.3 zeigt schematisch Safe-Action und Safe-Title.

¹Eingeblendete Erklärungen für das gezeigte Bild. Oft sind das Namen und Funktion von gezeigten Personen.

²Institut für Rundfunktechnik

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

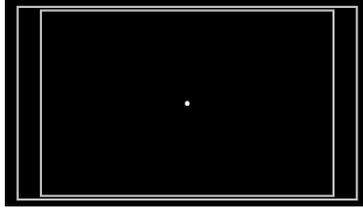


Abbildung 3.3: Safe-Title und Safe-Action im Videobild. Das innere Rechteck zeigt die Safe-Title-Begrenzung und damit den Bereich, der immer sichtbar ist. Das äußere Rechteck zeigt die Safe-Action-Grenze. Dieser Bereich ist meistens sichtbar und darf kurzzeitig verletzt werden. Logos werden normalerweise im Bereich des Safe-Titles platziert.

Das innere Rechteck ist dabei die Safe-Title-Vorgabe. Sie gibt einen optimalen Bereich für die Darstellung von Logos und Bauchbinden vor. Diese kennzeichnen die Marke einer Sendung und sollten deshalb immer im Bild sichtbar sein.

Safe-Action ist das äußere Rechteck. Es beschreibt die Begrenzung, in deren Rahmen Bewegungen ablaufen dürfen. Personen dürfen während einer Bewegung kurzzeitig im Fernsehbild angeschnitten dargestellt werden.

Obwohl diese technische Vorgabe für heutige Fernseher an Relevanz verliert, wird diese Begrenzung immer noch gern verwendet, weil sie eine Möglichkeit zur Orientierung im Videobild liefert. Durch die fest definierten Grenzen im Bild, können Objekte und Moderatoren an *Safe-Title* und *Safe-Action* besonders gut ausgerichtet werden. Das erleichtert die richtige Kadrierung von Einstellungen.

3.1.5 Kamerabewegungen und Kamerafahrten

„Es gibt viele Gründe den Blickwinkel der Kamera zu ändern: um einem Bildobjekt zu folgen, um wichtige Informationen zu geben oder vorzuenthalten, um die Erzählperspektive zu wechseln, um grafisch abwechslungsreich zu sein, einen Schauplatz vorzustellen oder eine Stimmung hervorzurufen“ [Kat98].

In der Filmbranche wird zwischen drei Arten von Kamerabewegungen unterschieden: Schwenk, Fahrt und Kranaufnahme. Die einzelnen Arten werden dabei häufig kombiniert. Meist erfordert eine Kamerafahrt auch einen Schwenk, um ein statisches Objekt während der Fahrt weiter im Bild zu halten [Kat98].

Kamera-Schwenks ändern die Orientierung der Kamera (Pan, Tilt). Kurze Schwenks führen oft von einem Aufmerksamkeitszentrum zum nächsten oder stellen eine visuelle Verbindung zwischen zwei oder mehreren Akteuren her [Ari00]. Schwenks mit langer Brennweite erscheinen

3.1 Theoretische Aspekte der Kameraführung

schneller, weil der kleinere Bildausschnitt sich schneller durch das Bild bewegt, als dies bei weitwinkligen Einstellungen der Fall ist.

Kamera-Fahrten verändern nach [Kat98] die horizontale Position der Kamera. Sie generieren beim Betrachter den Eindruck, als würde man sich durch den Raum bewegen. Unter einer Fahrt wird hier sowohl die tatsächliche Bewegung der Kamera als auch eine Zoombewegung bei stehender Kamera bezeichnet. Eine Bewegung zum Objekt hin und eine Zoomfahrt haben die gleiche Wirkrichtung. Sie bringen den Betrachter näher an die dargestellte Szene heran. Die Wirkung der Fahrten ist jedoch unterschiedlich. Abbildung 3.4 zeigt unter (a) die ursprüngliche *Totale*. 3.4 (b) zeigt das Resultat einer Zoomfahrt und 3.4 (c) zeigt eine Heranfahrt an die Szene. (b) und (c) zeigen eine vergleichbare Größe des Protagonisten. Dennoch verstärkt die Heranfahrt die Präsenz des Protagonisten im Vergleich zu einer Zoomfahrt. Wird die Kamera auf einen Akteur oder Objekt zubewegt, so wachsen der Kamera näher positionierte Objekte scheinbar schneller als Objekte im Hintergrund. Bei einer Änderung der Brennweite wachsen alle Objekte im selben Verhältnis [Ari00]. Deshalb werden gewöhnlich Heranfahrten der Zoomfahrt vorgezogen.



(a) Aufnahme in der *Totale*. (b) Gleicher Aufnahme- (c) Vergleichbare Größe der Standort. Die Szene wurde Hauptperson. Dabei wurde die über die Brennweite herange- Kamera näher an die Szene geholt ("Zoomfahrt"). gefahren.

Abbildung 3.4: Vergleich von *Totale-Einstellungen* die in (b) durch Zoom und in (c) durch eine Fahrt eingestellt wurden.

Nachrichten nehmen bei Kamerafahrten eine Sonderrolle ein. Rein dramaturgisch gibt es bei Nachrichten keine Emotion, die die Kamerafahrt unterstützen könnte. Der beginnende Trend zu mehr Bewegung bei Nachrichten lockert die Sendung auf und macht die Erscheinung weniger formal und diszipliniert. Ein weiterer Grund für lebhaftere Nachrichten ist, dass die heutige Technik dynamischere Präsentationsformen begünstigt. Neben den unbegrenzten Möglichkeiten der virtuellen Studioteknik, erlauben die aktuellen Kamerasysteme inzwischen reproduzierbare Aktionen, so dass Kamerabewegungen mit visuellen Effekten kombiniert werden können. Der Kameraschwenk auf einen Beitrag der *ZDF heute*-Nachrichten ist ein Beispiel dafür.

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

Bei Nachrichtensendungen bezieht sich eine Heran- bzw. Herausfahrt auf eine Fahrt von einer *Totalen* auf ein bestimmtes Objekt (OiF - Objekt im Fokus) zu oder umgekehrt. Die strenge Unterteilung des Filmbereichs gilt hier nicht. Jede Kamerabewegung kann als Fahrt bezeichnet werden. Eine Heran- bzw. Herausfahrt beginnt oft hoch oder tief und endet auf einer neutralen Endposition. Sie dient dazu, dem Zuschauer erst eine Übersicht über die Szene zu bieten, um dann den Schwerpunkt auf das OiF zu lenken. Das OiF ist bei Nachrichten der Moderator. Diese Form von Fahrt wird gern für den Ein- bzw. Ausstieg aus einer Sendung verwendet. Die wichtigsten Eigenschaften sind im Folgenden aufgezählt:

- Zu Beginn ist das OiF klein und unauffällig im Bild, am Ende der Fahrt kommt das OiF zu seiner Präsenz. Bei Nachrichten endet eine solche Fahrt meist in einer *Halbnahen* bzw. *Nahen*. Allgemein endet eine solche Fahrt auf dem OiF.
- Ein *Schwimmen* des OiF ist zu vermeiden. Unter *Schwimmen* wird hier verstanden, dass sich ein Objekt während der Fahrt horizontal hin und her bewegt. Eine solche Fahrt wirkt fehlerhaft und unharmonisch.
- Ein *Pumpen* des OiF ist zu vermeiden. Das Objekt verändert hierbei oszillierend die vertikale Position in der Bildebene. Oft entsteht das *Pumpen*, wenn der Zoom geändert wird. Bei handgeführten Kameras lässt sich dieser Effekt oft nicht vermeiden. Bei schnellen Fahrten wird dieser Effekt verstärkt. Handgeführte Kameras werden deshalb langsam bewegt.

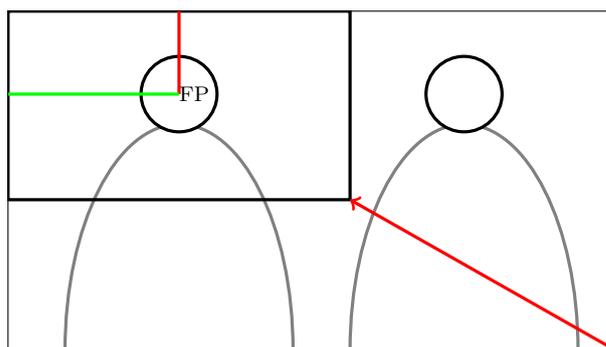


Abbildung 3.5: Schematische Darstellung eines Fahrtverlaufs von einer *Zweier-Einstellung* auf eine (zentrale) *Nahe*. Die Höhe und der Abstand von der rechten Seite des Kopfes ändern sich während der Fahrt nicht (Fixpunkt - FP).

Für die Fahrt aus einer *Zweier* in eine *Halbnahe* oder *Nahe* hält die Heranfahrt einen Fixpunkt (FP) im Bild. Sie soll so ausgeführt werden, dass die Position des im Endbild zu

sehenden Moderators bezogen auf den oberen Rand und dem, dem Moderator zugewandten Rand konstant bleibt. Abbildung 3.5 zeigt diese Eigenschaft schematisch für eine Fahrt aus einer *Zweier-Einstellung* in eine *Nahe-Einstellung*.

3.2 Praktische Aspekte der Kameraführung aus Sicht der Regie

In der Planungsphase einer Sendung wird ein sogenannter Einstellungskatalog (Storybook) erstellt. In diesem werden grob die für die Sendung gültigen Aufnahmepositionen und Einstellungen festgehalten. Dieser Katalog wird daraufhin im Studio verifiziert. Die Einstellungen werden in der realen Dekoration ausprobiert und diskutiert. Im weiteren Verlauf dieser Arbeit wird dieser Prozess *Einstellungsfindung* genannt. Überzeugen die gefundenen Einstellungen den Regisseur und den Kameramann, werden diese festgeschrieben und, soweit möglich, für die einzelnen Kameras am Boden markiert. Für die Lichtgestaltung sind die Aufnahmepositionen der Kamera sowie die Position der Moderatoren und evtl. Gäste verbindlich, denn nach diesen werden die Lampen im Studio ausgerichtet. Während der Sendung ist man an die eingeleuchteten Positionen im Studio gebunden. Für den regulären Sendebetrieb gelten diese Einstellungen dann als Vorgaben. Vor jeder Sendung werden diese Vorgaben an die aktuelle Situation angepasst. So müssen die einzelnen Einstellungen beispielsweise an die Größe des Moderators oder an nicht richtig positionierte Kulissenelemente angepasst werden. Diese kleinen Korrekturen an der Vorgabe werden hier als *Sendekontrolle* bezeichnet.

3.2.1 Kommunikation zwischen Regie und Kamera

In einem Fernsehstudio besteht oft kein Sichtkontakt zwischen Regie und Studio. Deshalb kommunizieren Studiopersonal und Regiepersonal über einen Studio-Funk, der ähnlich wie ein fest installiertes CB-Funkgerät funktioniert. Jeder Mitarbeiter des Studios hat einen Empfänger und kann so Anweisungen aus der Regie entgegennehmen. Der Regisseur lenkt die Kamera verbal. Dabei kann er sowohl festlegen, was gezeigt werden soll, als auch wie es zu zeigen ist. Grundlage dieser Kommunikation ist ein vorab besprochener Ablaufplan, in dem bestimmte Einstellungen den verschiedenen Themenblöcken der Sendung zugeordnet wurden.

Im Folgenden wird beispielhaft eine Kommunikationsszene vorgestellt. Diese wurde bei einem Besuch einer Magazinsendung aufgenommen und ist hier abgebildet und entsprechend kommentiert. Alle aufgeführten Kommandos sind von der Regie an eine Kamera gerichtet. Es

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

wird hier nur die Kommunikation zwischen Regie und Kamera1 wiedergegeben. Nachdem auf einem Set meist zwei oder mehr Kameras eingesetzt werden, wird hier auch die Adressierung der jeweiligen Kamera explizit mit angegeben. Kamera1 zeigt zu Beginn eine *Totale-Einstellung*.

"Kamera1 Achtung."

Es wird auf eine Aktion in naher Zukunft hingewiesen. Mit Hilfe dieser Anweisung weiß der Kameramann von Kamera1, dass er in naher Zukunft agieren muss und kann sich so schon einmal im Set orientieren und die nächste Bewegung vorplanen.

"Kamera1 zeig' mir eine Halb-Nahe."

Diese Anweisung fordert Kamera1 auf, jetzt eine *Halbnahe-Einstellung* einzunehmen.

"Kamera1 geh' ein wenig näher ran."

Die gezeigte Einstellung ist der Regie nicht nah genug. Sie fordert deshalb die Kamera auf, näher an den Moderator heranzufahren, um so den Moderator größer und damit präsenter im Bild zu zeigen.

"Kamera1 Gut so."

Die durchgeführte Änderung war ausreichend. Da die Änderung sehr langsam durchgeführt wurde, konnte die Regie den durchgeführten Änderungsvorgang beim Erreichen des gewünschten Ziels beenden.

"Kamera1 zeig mir jetzt die Nahe."

Hier wird ohne Schnitt von der *Halbnahen* auf eine *Nahe* gewechselt, das heißt der Kameramann verengt den Bildausschnitt so, dass der Moderator von Scheitel bis unter die Schultern gezeigt wird.

"Kamera1 geh' etwas höher, bis die Monitorecke aus dem Bild ist."

Die gezeigte Einstellung wird hier verbessert. Der Regie ist aufgefallen, dass am rechten unteren Bildrand ein Monitor der Studiodekoration angeschnitten zu sehen ist. Deshalb ergeht die Anweisung, dass die Einstellung zu ändern ist, bis dieser Monitor nicht mehr zu sehen ist.

"Kamera1 geh' zurück auf die Halbnahe"

Der Wechsel zurück auf die *Halbnahe-Einstellung*.

"Kamera1 Danke"

3.2 Praktische Aspekte der Kameraführung aus Sicht der Regie

In diesem Fall ist die Sendung hier beendet. Theoretisch könnte jetzt auf eine andere Kamera umgeschnitten werden oder ein Video-Einspieler erlaubt es, die Kamera1 offAir¹ umzupositionieren.

Zusammenfassend kann festgehalten werden, dass für die *Sendekontrolle* Kommandos zur Anfahrt bestimmter Positionen, Kommandos zum Umpositionieren der Kamera, Stopp-Kommandos und Konditionalkommandos während einer Sendung verwendet werden. Beim Umpositionieren wird ebenfalls eine Quantität mit übergeben. Hier wird weniger bzw. mehr Aktion erwartet, als wenn keine Quantität genannt wäre. Für die *Einstellungsfindung* ist davon auszugehen, dass diese Kommandos ebenfalls verwendet werden. Sie werden ergänzt durch Kommandos zum Speichern der aktuellen Position.

Bei keinem der Kommandos wurde eine Bestätigung seitens der Kamera gegeben. Dieser Umstand liegt darin begründet, dass die Regie das Ergebnis der Kameraführung auf einem Kontrollmonitor verfolgen kann und dass das Studiopersonal angehalten ist, während der Sendung nicht zu sprechen. Der Moderator würde sonst abgelenkt werden.

Ferner fällt auf, dass die Kommandos im Indikativ formuliert werden. Die besuchte Sendung war eine Live-Magazinsendung. In dieser Situation ist für Erklärungen keine Zeit. Existiert ein Problem, so wird dieses direkt angesprochen.

3.2.2 Ablauf einer Sendung am Beispiel von *Zur Sache Baden-Württemberg*

Im Regelbetrieb wird für jeden Abschnitt der Sendung ein sogenannter Sendeplan ausgearbeitet. Dieser beschreibt, welche Kamera welche Einstellungen während der Sendung übernimmt. Gibt es mehrere Kameras, wird eine Kamera ausgewählt, die ständig den Moderator anvisiert sowie andere, die z.B. Detailaufnahmen von Requisiten oder der Gäste sowie *Totale-Einstellungen* des Studios zeigen.

Abbildung 3.6 zeigt das Ergebnis eines solchen Sendeplans für *Zur Sache Baden-Württemberg* vom 22.09.2011. Bei Sendungseröffnung zeigt Kamera2 einen Stoff-Bären in einer *Nahen*. Daraufhin wird die Einstellung auf eine *Halbnahe* und schließlich auf eine *Zweier-Einstellung* vergrößert (Einstellungsdefinition siehe Kapitel 3.1.2). Kamera3 beginnt mit dem Zeigen des Bärs, um dann auch in eine *Nahe-Einstellung* des Moderators zu wechseln. Kamera1 zeigt immer den Gast in einer *Nahen-Einstellung*. Die Sendungseröffnung schließt schließlich wieder

¹offAir bezeichnet einen Zustand der Kamera, bei dem die aufgenommenen Bilder nicht am Regieausgang anliegen.

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

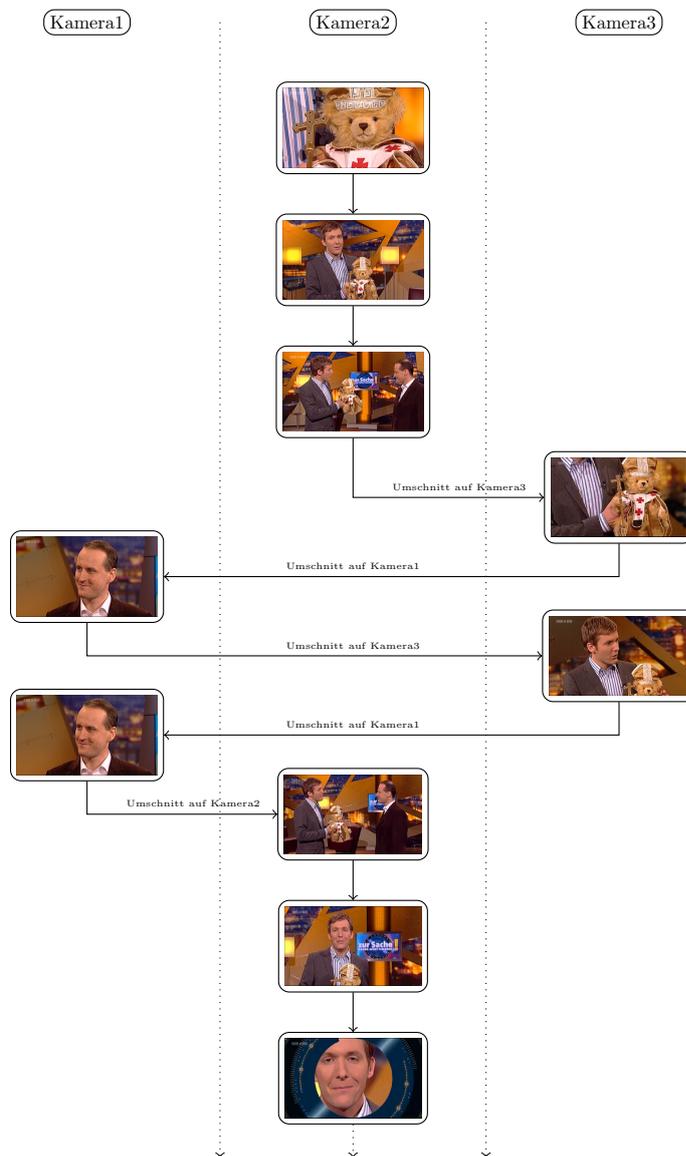


Abbildung 3.6: Zuordnung Einstellungsmotiv und Kamera der Anfangssequenz von *Zur Sache Baden-Württemberg* am 22.09.2011. Kamera1 zeigt in dieser Sequenz immer eine Nahe des Gasts. Kamera2 und Kamera3 wechseln zwischen unterschiedlichen Einstellungen und Motiven.

3.2 Praktische Aspekte der Kameraführung aus Sicht der Regie

mit Kamera2, die von einer *Zweier-Einstellung* über eine *Halbnahe* zu einer *Nahen* wechselt. Daraufhin erfolgt der Umschnitt auf einen Sendebbeitrag.

Für eine Sendung gilt, dass die einzelnen Einstellungen grob vorgegeben werden und der Kameramann entscheidet, wie er die Anweisungen interpretiert und folglich, wie er das geforderte Bild einstellt. Kamera1 zeigt in diesem Abschnitt immer den Gast in einer *Nahen-Einstellung*. Sie muss dabei auf Bewegungen des Gasts reagieren.

3.2.3 Ergebnisse der Feldstudie zur praktischen Kameraarbeit

In der Literatur gibt es unzählige Hinweise und Tipps darauf, wie eine Kamera geführt werden sollte. Dort werden verschiedene Einstellungen und deren Wirkung beschrieben. Dabei geht es immer um die Aussagekraft des jeweiligen Bildes und um die Aussage, die der jeweilig Verantwortliche dem Bild mitgeben will. Hier werden qualitative Aussagen gefällt, die für den szenischen Betrieb gelten. Bei Standard-Sendungen, die regelmäßig produziert werden, können diese Aussagen nur bedingt angewandt werden. Deshalb haben sich im Nachrichten und Magazinbereich bestimmte Regeln etabliert, die in der Praxis als zweckmäßig erachtet werden. Bilder, die nach Lehrbuchauffassung durchaus richtig gestaltet wurden, können für einen Regisseur im Nachrichtenbereich undenkbar sein. Oft geht es um die kleinen Dinge an der Einstellung. Der Grundaufbau entspricht hier der Lehrbuchmeinung, aber dennoch gibt es unzählige Möglichkeiten die gleiche Einstellung wiederzugeben. Die Entscheidung hierüber fällen Kameramann und Regisseur. Die Literatur kann keine Aussagen darüber fällen, welche Bildeigenschaften bei Nachrichten bzw. Magazinen tatsächlich relevant sind und in der Praxis eingefordert werden und welche nicht.

Ziel eines intelligenten, automatischen Kamerasystems¹ (IAK) ist es, sich so zu verhalten, wie es eine, durch einen Kameramann bzw. -frau geführte, Kamera tun würde. Aus diesem Grund wurden verschiedene Interviews mit unterschiedlichen Berufsgruppen aus dem Studiobereich geführt. Für ein IAK wird es entscheidend sein, neben dem expliziten Wissen über Kameraführung auch das implizite Wissen der Kameramänner und des Studiopersonals zu berücksichtigen. Sicher können aufgeführte Interviews keinen Anspruch auf Vollzählig- und Allgemeingültigkeit erheben, aber dennoch bieten sie eine gute Sammlung an individuellem Wissen, auf das weiter aufgebaut werden kann.

¹Im Weiteren werden die Begriffe intelligentes, automatische Kamerasystem und intelligent gesteuertes Kamerasystem synonym verwendet.

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

Neben der Einstellung selbst ist auch die Kommunikationsform zwischen Regie und Kamera von Bedeutung. Es hat sich im Studiobereich eine eigene Sprache etabliert, um bestimmte Anweisungen von der Regie an die Kamera weiterzugeben. Ein IAK muss in der Lage sein, diese Sprache zu verstehen und sich entsprechend zu verhalten. Deshalb wurden in den Interviews ebenfalls typische Kommandos und die typische Kommunikation zwischen Regie und Kamera thematisiert. Es wurden insgesamt elf Personen aus verschiedenen Berufsgruppen, die im Bereich TV-Studio, Nachrichten- und Magazinsbereich tätig sind, befragt. Die Interviews wurden meist während des Live-Betriebes geführt, stichpunktartig mitgeschrieben und später ausformuliert. Im Folgenden werden die für diese Arbeit relevanten Aspekte der Feldstudie zusammengefasst vorgestellt.

3.2.3.1 Typische Anweisungen aus der Regie

Der Regisseur hat oft einen besseren Eindruck vom Gesamtbild der Sendung und kann so, bei Bedarf, steuernd eingreifen. Häufige Anweisungen aus der Regie sind z.B. „Geh näher ran“ oder „Setz’ den Moderator mehr links/rechts“. Oft werden auch störende Elemente im Bild angesprochen. Sind mehrere Kameras im Einsatz, wird eine bestimmte Kamera direkt angesprochen, z.B. „Die Zwei Halb-Nahe“. Diese Anweisung gilt für Kamera2 und wünscht eine *Halbnahe-Einstellung*. Anweisungen aus der Regie können und müssen möglichst kurz und eindeutig gehalten werden, weil während einer Sendung keine Zeit für lange Erklärungen oder Beschreibungen ist. Durch feste Zuständigkeiten, meist langjährige Zusammenarbeit und übereinstimmendes Bildverständnis ist allen Beteiligten auch bei sehr kurzen Anweisungen klar, was gemeint ist und wie zu reagieren ist. Alle Anweisungen beziehen sich immer auf das gezeigte oder das gewünschte Bild. Alle Angaben sind immer aus der Sicht des Betrachters gegeben. Für ihn ist die Bewegung des Bildinhalts ausschlaggebend. Die tatsächliche Bewegungsrichtung ist nicht interessant.

In einem Studio herrscht unausgesprochen eine Übereinkunft, welche Begriffe für welche Einstellungen gelten. Die in Kapitel 3.1.2 vorgestellten Einstellungsgrößen dienen hier als Basis. Für jedes Studio existieren spezielle Anpassungen, die angeben wie weit die einzelnen Größen gelten. So sprechen die Befragten beim SWR bei Nachrichten von *Nahe-*, *halbtotale-* und *totaler Einstellung*. Diese Unterteilung entspricht der *Nahen*, *Halbnahen* und *Totalen-Einstellung* beim ZDF und aus Kapitel 3.1.2. Der Kameramann weiß, dass er bei „Geh etwas weiter weg“ mehr von der Studioumgebung zeigen muss. Um wie viel mehr geht einerseits aus seinem Bildverständnis,

3.2 Praktische Aspekte der Kameraführung aus Sicht der Regie

den Erfahrungen mit dem jeweiligen Regisseur oder aber durch direkte Anweisungen in Form von *Weiter* und *Stopp* hervor.

Vorrangig werden Bild- bzw. Einstellungskorrekturen aus der Regie heraus erteilt. Die grobe Einteilung, welche Kamera wann welches Objekt wie zeigen soll, wurde bereits bei der Regiebesprechung vor der Sendung festgelegt. Jeder Kameramann weiß, wo er sich bewegen kann und welche Einstellung er zeigen soll. Fällt dem Regisseur dennoch etwas während der Sendung auf, weist er aus der Regie auf diese Auffälligkeit hin und korrigiert das Bild verbal. Die einzelnen Kameras bieten die jeweils besprochenen Bilder an. Die Anweisungen aus der Regie weisen auf Fehler hin und bieten Vorschläge diese zu lösen. Die Anweisungen dienen also der Kontrolle, sodass die gezeigten Bilder einer Sendung zusammenpassen.

3.2.3.2 Merkmale eines guten Bildes

Bei der Frage nach einem guten Bild stößt man immer wieder auf die Begriffe *Goldener Schnitt* und Kadrierung. Eine eindeutige Antwort auf diese Frage nach dem Schönen und Guten konnten auch die befragten Interview-Partner nicht geben, weil ein Bild von jedem Menschen subjektiv anders wahrgenommen wird. Aus den Interviews lassen sich aber einige Richtlinien ableiten, die ein guter Kameramann beachtet und damit auch von einem automatischen Kamerasystem beachtet werden sollten.

Wichtig für die richtige Einstellung einer Szene, ist der Eindruck, den ein Bild transportieren soll. An dieser Stelle unterscheiden sich Nachrichten- und Magazinsendungen von szenischen Produktionen.

Bei szenischen Produktionen sollte die Kameraführung zu der Stimmung in der jeweiligen Szene passen. Das heißt in einer spannenden Szene sollte die Kamera diese Spannung z.B. durch späte relativ schnelle Bewegungen unterstützen, während in melancholischen Szenen die Kamera behutsam, flüssig und eher langsam bewegt werden sollte. Der Bildinhalt bewegt sich hier in einem breiten Spektrum. Angefangen bei weiten Aufnahmen, die dem Zuschauer über die Situation in der Szene in Kenntnis setzen, bis hin zur Super-Nah-Aufnahme bei der z.B. das „Entsetzen“ der Hauptperson gezeigt wird, ist hier alles erlaubt, was gefällt. Ein Patentrezept für die Umsetzung von Szenen kann in diesem Fall nicht gegeben werden. Dafür ist die künstlerische Freiheit zu groß und oft wirken ungewöhnliche Einstellungen am besten.

Im Informationsbereich gelten andere Regeln. Hier ist bei der Präsentation der Nachrichten in der Regel kein Platz für Emotionen. Die einzelnen Meldungen sollten sachlich und distanziert vorgestellt werden. Der Zuschauer darf keine Sympathie oder Antipathie für eine bestimmte

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

Meldung spüren. Nachrichten sollen unabhängig und unvoreingenommen präsentiert werden. Diese Grundregel beeinflusst die Führung der Kamera. Prinzipiell wird die Kamera bei Nachrichten in einer neutralen Position gehalten. Das heißt, sie ist weder auf- noch untersichtig. Bei Moderatoren unterschiedlicher Körpergröße darf von dieser Grundregel minimal abgewichen werden. Falls möglich wird eine Höhenänderung jedoch immer der Neigung vorgezogen.

Mit neutraler Beobachter-Position lassen sich Nachrichten besser darstellen als mit *Nahe-Einstellungen*, die oft emotional wirken. Auf den *Goldenen Schnitt* wird besonders geachtet. Das bedeutet, die Augen des Moderators sollten auf dem Schnittpunkt der linken oder rechten vertikalen und insbesondere der oberen horizontalen Linie des *Goldenen Schnitts* liegen. Dabei ist auch darauf zu achten, dass nach oben ausreichend optischer Raum vorhanden ist. Dieser Raum wird als *Luft* bezeichnet.

Sollte ein Umschnitt erfolgen, wird darauf geachtet, dass der Moderator sowohl vor, als auch nach dem Schnitt auf der gleichen Seite im Bild zu sehen ist. Ein Positionswechsel des Moderators im Bild ist während einer Moderation zu vermeiden.

Ein weiteres Merkmal ist die *Luft* über dem Kopf. Fast alle Interviewpartner geben hier an, dass diese Größe bei Nachrichten eine wichtige Eigenschaft des Bildes ist. Die *Luft* ist bei Nachrichten fest vorgegeben und darf, im Gegensatz zu Magazinsendungen nicht verändert werden. Bei Magazinsendungen ist ein Anschnitt des Kopfes dagegen „sendefähig“.

3.2.3.3 Merkmale eines guten Kameramanns

Kameraführung ist ein kreativer Prozess, der als primäres Ziel die Darstellung einer bestimmten Szene hat. Vor diesem Hintergrund gaben die Interview-Partner an, dass ein Kameramann den Weg zwischen zuschaueransprechenden Bildern, Weiterführung der erzählten Geschichte sowie Einschränkungen durch das Genre bzw. durch die Vorgaben des Senders finden muss. Unterstützt bzw. kontrolliert wird dieser Prozess durch den Regisseur, der die Gesamtverantwortung über alle produzierten Bilder trägt.

Ein guter Kameramann liefert aufgrund seiner Erfahrung gute Bilder, darin sind sich die befragten Personen einig. Der Begriff „gutes Bild“ hängt stark vom Zweck der Aufnahme ab. Bei einer szenischen Produktion muss ein gutes Bild zur aktuellen Szene passen. Die Stimmung innerhalb der Szene wird dabei durch die Kamera (bzw. Kameraführung) bestmöglich unterstützt. In diesem Fall sollte der Kameramann verstehen, wie die Szene in Bildern dargestellt werden kann und für welche Effekte welche Bildeigenschaften gut und wichtig sind.

3.2 Praktische Aspekte der Kameraführung aus Sicht der Regie

Für eine Nachrichten- bzw. Magazinproduktion gibt es feste Vorgaben, die eingehalten werden müssen. Die Vorgaben wurden einmal in einem Storybook festgelegt. Für jede Sendung werden die Vorgaben wieder eingestellt und repräsentieren die Ästhetik der Sendung. Die Qualität eines Kameramanns misst sich hier daran, inwieweit er „schöne“ Bilder innerhalb der Vorgaben generiert. So gilt es beispielsweise die *Luft über dem Kopf* auch während einer Ranfahrt zu halten, gewünschte Proportionen auch bei mehreren Objekten im Bild auszugleichen oder unerwünschte Effekte im Zusammenhang mit der Dekoration zu vermeiden. Außerdem müssen die Vorgaben auf den jeweiligen Moderator angepasst werden. Dafür wird unter Beibehaltung der Perspektive durch Pan, Tilt und Zoom das Bild den Vorgaben entsprechend angepasst.

Bei Live-Sendungen ist für einen sicheren Sendungsverlauf wichtig, dass der Kameramann stets den Überblick bewahrt. Unerwartete Situationen müssen professionell abgehandelt werden. In diesem Fall weiß ein guter Kameramann auch ohne Absprache, wie Bilder aussehen müssen. Steht z.B. bei einer Show ein Gast nicht dort, wo er während der Probe stand, schafft es ein guter Kameramann trotzdem „gute Bilder“ zu liefern. Gleiches gilt, wenn z.B. ein Gast nicht den vorgesehen Weg geht oder beispielsweise in einer Talk-Sendung „vor Empörung aufspringt und die Sendung verlässt“. Ein guter Kameramann versteht es, in dieser Situation dem Zuschauer klare und verständliche Bilder zu liefern.

Ein guter Kameramann sollte die Kamera handwerklich beherrschen und die von der Regie gewünschten Einstellung richtig umsetzen können. Die Regie wacht über die Zusammenstellung der einzelnen Kamerabilder. Deshalb darf und muss sie Korrekturen vornehmen können, um dem Zuschauer einen einheitlichen Bildeindruck zu präsentieren. Bei einem guten Kameramann sind nicht viele Worte nötig. Er weiß wie Bilder in dem jeweiligen Fall auszusehen haben und gleicht sich, bei Bedarf, automatisch an die vorhergehenden Bilder an.

3.2.3.4 Genre-spezifische Merkmale

Zunächst muss festgehalten werden, dass bei den meisten Fernsehproduktionen weitgehend harmonisch kadriert wird. Um den Zuschauer nicht mit ungewöhnlichen Einstellungen zu überraschen, müssen ihm stets gewohnte Bilder geliefert werden. Anders ist das im Experimental- und Kurzfilm- sowie im Kino-Bereich. Hier kann und soll bewusst ungewöhnlich kadriert werden, um dem Zuschauer ein besonderes Filmerlebnis zu bieten.

Bei Nachrichtensendungen muss die Kamera die Handlung harmonisch abbilden und durch eine neutrale Aufnahmeposition Seriosität ausstrahlen. Die Kamera befindet sich dabei auf

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

der Augenhöhe des Moderators. Spektakuläre Kamerafahrten sind selten, die Anzahl der Einstellungen ist begrenzt und ihre Ausprägung festgeschrieben. Für Nachrichten ist es wichtig, wiedererkennbar und abgrenzbar zu anderen Sendungen zu sein. Deshalb werden feste Vorgaben geschaffen, die in jeder Sendung möglichst genau nachgebildet werden.

Nachrichten bzw. Magazinsendungen dienen der Information des Zuschauers und zeigen drei wiederkehrende Elemente im Bild: Zum einen den Moderator, die Themenfläche und evtl. einen Gast oder Experten. Der Moderator präsentiert die einzelnen Themen und ist das „Gesicht der Sendung“. Die Themenfläche wird hinter oder seitlich vom Moderator platziert. Dort werden Bilder zum aktuellen Beitrag projiziert. Der Gast wird in einer Interview-Situation zu einem bestimmten Thema befragt und ist sonst nicht zu sehen.

3.2.3.5 Merkmale einer guten Kamerafahrt

Es gibt verschiedene Kriterien für eine gute Kamerafahrt. Dem befragten Studiopersonal war besonders die ruhige Bewegung, der konstante Fokus auf das jeweilige Objekt sowie die dramaturgische Eingliederung der Fahrt in den Sendungsablauf wichtig. Für das befragte Personal war es außerdem wichtig, dass eine Kamerabewegung einen definierten Anfang und ein bestimmtes Ende besitzt. Der Zuschauer muss also feststellen können, wann eine Fahrt begonnen hat und wann diese beendet ist. Vor allem das Ende sollte dramaturgisch passen. D.h. mit dem Ende einer Moderation sollte auch eine Fahrt auf der gewünschten Einstellung zum Ende kommen. Fahrten sollten in diesem Genre immer von einer Einstellung auf eine andere führen.

Während einer Bewegung sollte die Bildaufteilung weiterhin stimmen. Verschiebt sich diese während der Fahrt, wirkt das für den Zuschauer befremdlich. Eine Fahrt durch leere Räume ist ebenso zu vermeiden. Bei solchen Bewegungen verliert der Zuschauer den Bezug zum Raum und wird so irritiert. Fahrten sollten deshalb möglichst offen gefahren werden. So hat der Zuschauer die Chance, seine Orientierung auch während der Fahrt zu behalten.

Zoomfahrten sind in szenischen Produktionen nicht gern gesehen. Die bessere Wahl ist es hier, die Kamera selbst zu bewegen. Da eine Bewegung der Kamera das Risiko birgt „Ruckler“ zu generieren, werden in Nachrichtensendungen Korrekturen vermehrt durch den Zoom ausgeführt und so das Risiko von „Rucklern“ im Bild ausgeschlossen.

3.3 Vergleich der Nachrichten von ARD, ZDF und RTL aus der Sicht des Zuschauers

Es gibt viele verschiedene Analysen über Nachrichten. Die meisten beziehen sich dabei allerdings darauf, wie hoch der jeweilige Anteil der Meldungen aus dem Bereich der Politik, Wissenschaft, Kultur oder ähnlichen ist oder sie beziehen sich auf die geographischen Gegebenheiten der Meldungen. Ferner kann der Textstil, die Präsentationsform, Musikanteil, räumliche Gestaltung, die Anteile der Hard -bzw. Softnews und die Verwendung von “Symbolen, Karten, Grafiken oder Kurven” [Fau08] analysiert werden.

Alle diese Analysen sind für das hier verfolgte Ziel wenig aussagekräftig. Sie beziehen sich auf die Arbeit der Redaktion und sind stark von deren Vorgaben abhängig. Im Folgenden werden Nachrichtensendungen von bekannten deutschen Sendern hinsichtlich der automatisierbaren Eigenschaften wie der Schnittfolge bzw. der Einstellungshäufigkeit sowie der Bildkomposition analysiert. Diese Kriterien zeigen den Ablauf von Nachrichten, die Verwendung und den Verlauf von Fahrten innerhalb der Sendung sowie die verwendeten Einstellungen und deren Bildkomposition. Die gewählten Nachrichten von ARD, ZDF und RTL sind die Einschaltquoten-stärksten in Deutschland. Die hier vorgestellten Bilder stammen alle aus Sendungen vom 18.02.2011.

3.3.1 Einstellungsfolge während einer Sendung

Das Erscheinungsbild einer Sendung wird entscheidend von der Einstellungsfolge geprägt. Im Folgenden werden die Einstellungsfolgen der drei untersuchten Sendungen grafisch aufgezeigt. Ziel dieses Kapitels ist es, die für Nachrichten verwendeten Einstellungen sowie deren Häufigkeit in einer Sendung vorzustellen. Ferner ist im Verlauf dieses Kapitels zu sehen, dass die einzelnen Meldungen durch unterschiedliche Einstellungen vorgetragen werden. Eine Nachrichtensendung muss nicht auf eine einzige Einstellung beschränkt sein.

Für die Graphen in den folgenden Abbildungen wurde die Einteilung der Einstellungsgrößen an das in Abschnitt 3.1.2 eingeführte Modell angelehnt. *Clip* steht dabei jeweils für einen Filmbeitrag der Nachrichten. Er wird im Graph wie eine Einstellung oberhalb der *Nahen* einsortiert. So können alle Übergänge innerhalb einer Sendung dargestellt werden. Bei der *tagesschau* wird ein drei-stufiges Modell mit den Einstellungsgrößen *Clip*, *Halbnahe* und *Totale* verwendet. Bei der *heute*-Sendung des ZDFs wird ein fünf-stufiges Modell, beginnend bei *Clip*, über *Nahe*, *Halbnahe*, *Zweier* und *Totale* verwendet. Bei *RTL Aktuell* musste das bei *ZDF heute* ange-

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

wandte Modell um eine *Amerikanische-Einstellung* erweitert werden. Sie liegt zwischen einer *Halbnahen* und einer *Zweier-Einstellung*.

Jede Einstellung ist mit der Einheit 1BE bezeichnet. Die Dauer der einzelnen Einstellungen kann hier nicht abgelesen werden. Fahrten werden durch Schrägen zwischen zwei Einstellungen und Umschnitte durch vertikale Graphverläufe dargestellt.

Die *tagesschau* ist unter den drei untersuchten die zuschauerstärkste Nachrichtensendung. Die Kamera bewegt sich während der Sendung nicht. Einspieler werden hart geschnitten, die Kamera zeigt fest eine *Halbnahe-Einstellung*. Das *Opening* enthält eine kurze Kamerafahrt aus der seitlichen *Totale* mit Schnitt in die *Halbnahe*. Abbildung 3.7 zeigt den Verlauf der *tagesschau* vom 18.02.2011. Die Sendung besteht aus einem Wechsel von Moderation und Einspielern.

Im Graph ist der Schnitt beim Wechsel zwischen Moderation und *Halbnahe* durch die vertikale Verbindung zu erkennen. Die Sendung zeigt abgesehen vom *Opening* und *Ending* immer eine *Halbnahe-Einstellung*.

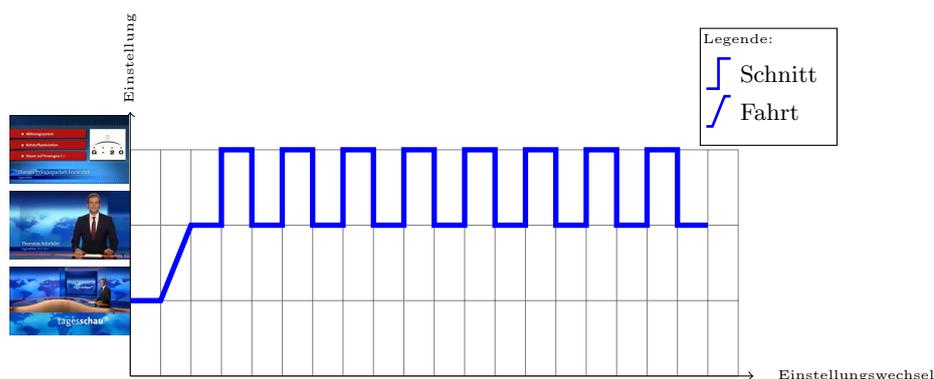


Abbildung 3.7: Einstellungs-Verlauf der *tagesschau*-Sendung am 18.02.2011. Eine kurze Fahrt wechselt von der *Totale* auf eine *Halbnahe*. Daraufhin werden die Einspieler hart geschnitten. Fahrten werden nach der Eröffnung nicht mehr gezeigt.

Die *ZDF heute*-Sendung ist seit 17.07.2009 mit dem "neuen Nachrichtenstudio" auf Sendung [SEB09]. Abbildung 3.8 sowie Abbildung 3.9 zeigen die Einstellungsfolge vom 18.02.2011. Im Vergleich mit der Darstellung der *tagesschau* (Abbildung 3.7) fällt die größere Zahl an Einstellungen sowie deren abwechslungsreichere Gestaltung auf. Das *Opening* ist das typische Merkmal der *heute*-Sendung. Hier wird zu Beginn die bekannte Uhr gezeigt, gefolgt von einer Fahrt aus der *Totale* in die *Nahe-Einstellung*. Nach der persönlichen Begrüßung durch den Moderator, erfolgt ein Schnitt auf den Themenüberblick. Nach dem Schnitt zurück in die *Halbnahe* folgen die einzelnen Meldungen und die dazugehörigen Berichterstattungen.

3.3 Vergleich der Nachrichten von ARD, ZDF und RTL aus der Sicht des Zuschauers

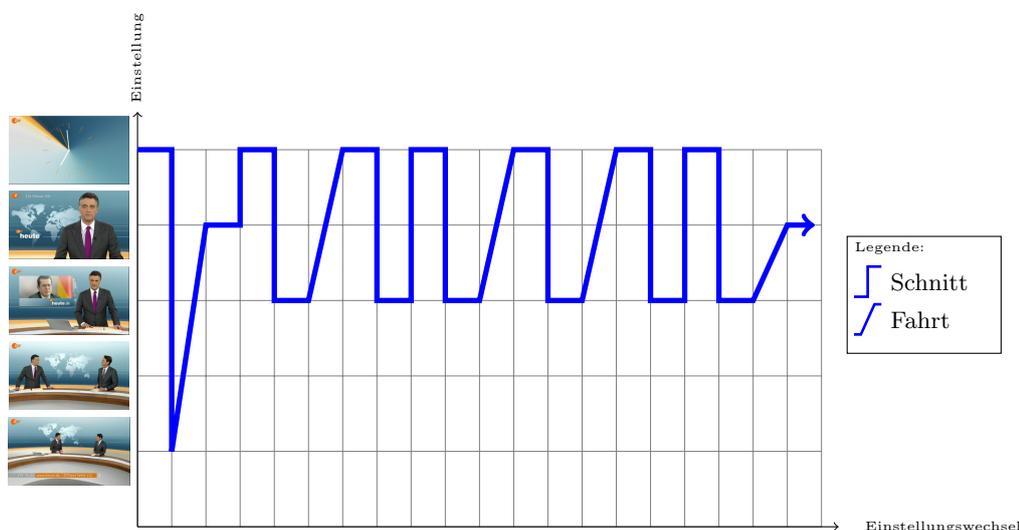


Abbildung 3.8: Einstellungs-Verlauf der *heute*-Sendung am 18.02.2011 - Teil 1. Die Haupteinstellung ist hier die *Halbnahe-Einstellung*. Diese wird durch verschiedene Fahrten in einen *Clip*, bzw. in eine andere Einstellung ergänzt.

Seit der Inbetriebnahme des neuen Nachrichtenstudios besteht die Möglichkeit einer virtuellen Kamerafahrt (Virtual Camera Flight – VCF) in den jeweiligen Clip hinein. Im Graph ist das durch die schräge Verbindung zwischen der *Halbnahen-Einstellung* und dem *Clip* dargestellt. Auf diese Weise hat das ZDF mehr Optionen, um in einen Beitrag zu schneiden und kann so die Sendung abwechslungsreicher gestalten.

Sport wird bei der *ZDF heute*-Sendung von einem Co-Moderator präsentiert. Die Übergabe zu diesem geschieht über eine *Zweier-Einstellung*. Der Sportmoderator wird gleich kadriert wie der Hauptmoderator. Er wird in einer *Nahen* oder *Halbnahen-Einstellung* gezeigt. Im Graph wird diese Einstellung nicht extra gekennzeichnet. Nach der Rückgabe an den Hauptmoderator folgt die Verabschiedung und das Ending.

Die Nachrichtensendung von RTL ist, von den hier ausgewählten Sendungen, die aktuellste und einstellungsreichste Sendung. Sie wurde am 11. September 2010 in dieser Form erstmals ausgestrahlt. Zusätzlich zu den von ZDF bekannten Einstellungen gibt es hier in jeder Sendung eine *Amerikanische-Einstellung*. Die Sendung eröffnet, ähnlich wie das ZDF, durch eine Grafik, die zu einer Fahrt umgeschnitten wird. Diese Fahrt endet in einer *Halbnahen-Einstellung* zur Begrüßung der Zuschauer. Die einzelnen Meldungen werden mit Hilfe von unterschiedlichen Einstellungen präsentiert. Bei dieser Sendung gibt es die Möglichkeit aus der *Halbnahen*, *Nahen*

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

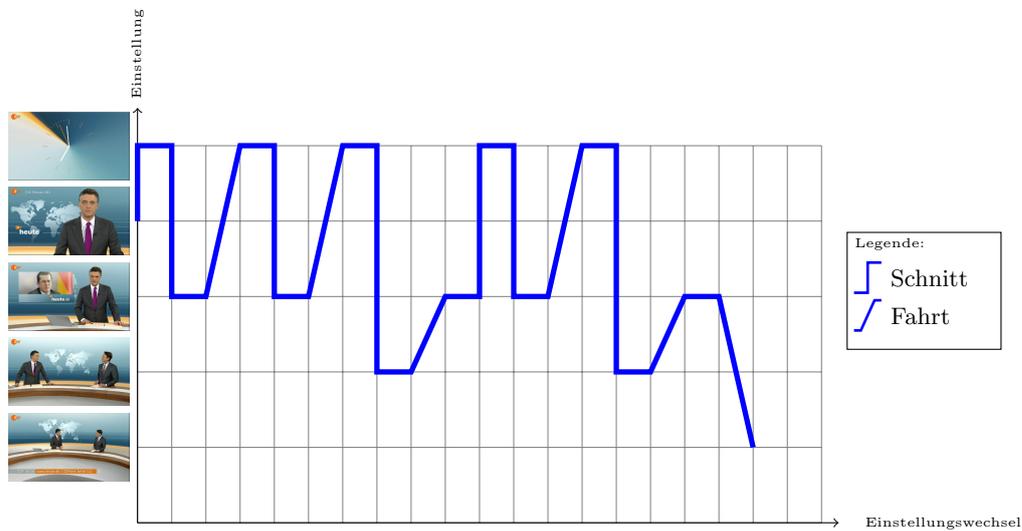


Abbildung 3.9: Einstellungs-Verlauf der *heute*-Sendung am 18.02.2011 - Teil 2. Fahrten werden auch im zweiten Teil der Sendung für Themenwechsel bzw. Fahrten in Clips verwendet.

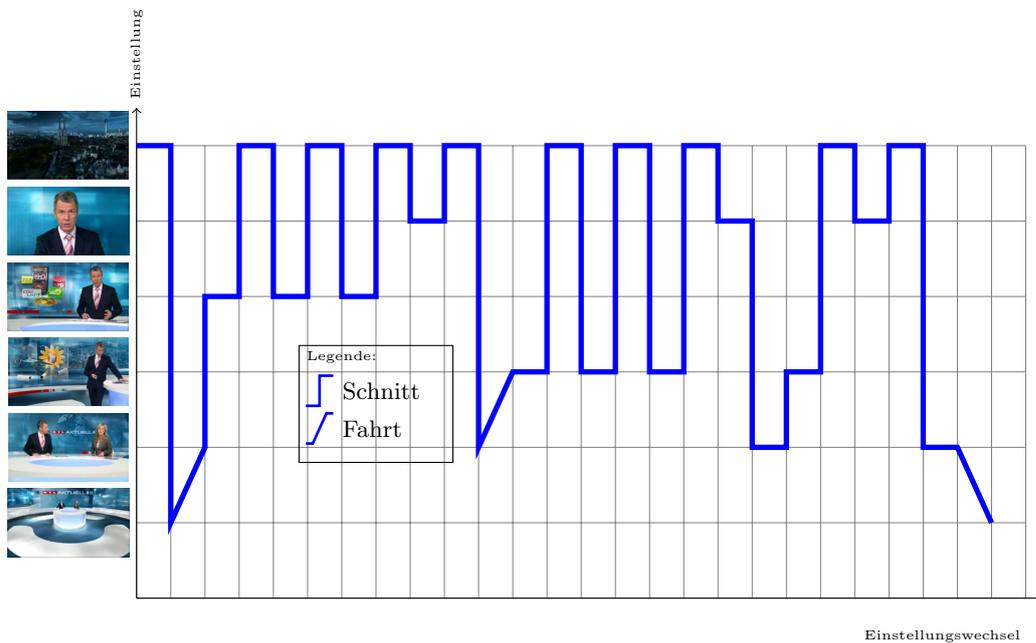


Abbildung 3.10: Einstellungs-Verlauf der *RTL Aktuell*-Sendung am 18.02.2011. Diese Sendung ist die einstellungsreichste Sendung unter den hier Vergleichenen. Fahrten existieren hauptsächlich als Übergang zwischen Einstellungen.

oder *Amerikanischen-Einstellung* Meldungen zu präsentieren. Auch hier gibt es eine *Zweier-*

Einstellung, die zur Übergabe von einem Moderator zum anderen verwendet wird.

3.3.2 Position und Größe des Hauptmoderators während einer Sendung

Die folgenden Abbildungen zeigen Einstellungen der betrachteten Sendungen. Diese wurden beliebig aus unterschiedlichen Sendungen ausgewählt und zeigen, nach Sender sortiert, Einstellungen der jeweiligen Sendung mit unterschiedlichen Moderatoren. Ziel dieses Kapitel ist es, Grundgrößen für bestimmte Einstellungen ableiten zu können.

Die dargestellten Linien sind grob in die jeweiligen Zeichnungen eingefügt worden. Sie geben jeweils die horizontale Position der Moderatoren, deren Kopfgröße und die *Luft* über dem Kopf des Moderators an. Die Einheit *Luft* ist im Sprachgebrauch der Fernsehproduktion eine durchaus geläufige Bezeichnung. Sie wird deshalb auch hier verwendet. Alle Größen sind relativ zur Bildbreite bzw. Bildhöhe angegeben. Sie wurden grob bestimmt und sind deshalb nur als Richtwert für die jeweiligen Einstellungen zu verstehen.

Die Abbildungen 3.11 (a) und (b) zeigen *Nahe-Einstellungen* von *RTL Aktuell* und die Abbildungen 3.11 (c) und (d) die von *ZDF heute*. In der *tagesschau* kommen keine *Nahe-Einstellungen* vor. Die dargestellten Bilder zeigen jeweils zwei *Nahe-Einstellungen* mit unterschiedlichen Moderatoren. Die zwei Bilder von *RTL Aktuell* unterscheiden sich hinsichtlich Positionen und Dimensionen im Bild nicht. Beide Moderatoren sind auf eine Position von ca. 30% der Bildbreite gesetzt und haben über dem Kopf einen Abstand von ca. 7% der Bildhöhe. Dieser Abstand wird im Film- und Fernsehbereich als *Luft* bezeichnet. Die Größe der Moderatoren unterscheidet sich ebenfalls nicht. Hier wird die Darstellungsgröße der Moderatoren durch die vertikale Ausdehnung des Kopfes im Bild bestimmt. Für beide Bilder nimmt die Höhe des Kopfes ca. 42% der Bildhöhe ein.

Die Abbildungen 3.11 (c) und (d) zeigen die *Nahe-Einstellungen* der *heute*-Nachrichten des ZDFs ebenfalls für zwei willkürlich ausgewählte Moderatoren. Die Moderatoren sind bei diesen Einstellungen auf der rechten Seite auf ca. 71% der Bildbreite positioniert. Für die unter (c) dargestellte Abbildung sind hier wieder ca. 7% der Bildhöhe für die *Luft* über dem Kopf reserviert. In Abbildung (d) sind es hingegen 2%. Die Moderatorengrößen sind jeweils in ähnlicher Größenordnung wie bei den Nachrichten von RTL. Für die linke Abbildung liegt sie bei ca. 42% für die rechte liegt sie bei ca. 40% der Bildhöhe.

Für die *Halbnahe-Einstellung* liegen Beispiel-Bilder von allen drei hier verglichenen Nachrichtensendungen vor. In Abbildung 3.12 (a) und (b) sind zwei Einstellungen der ARD *tages-*

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

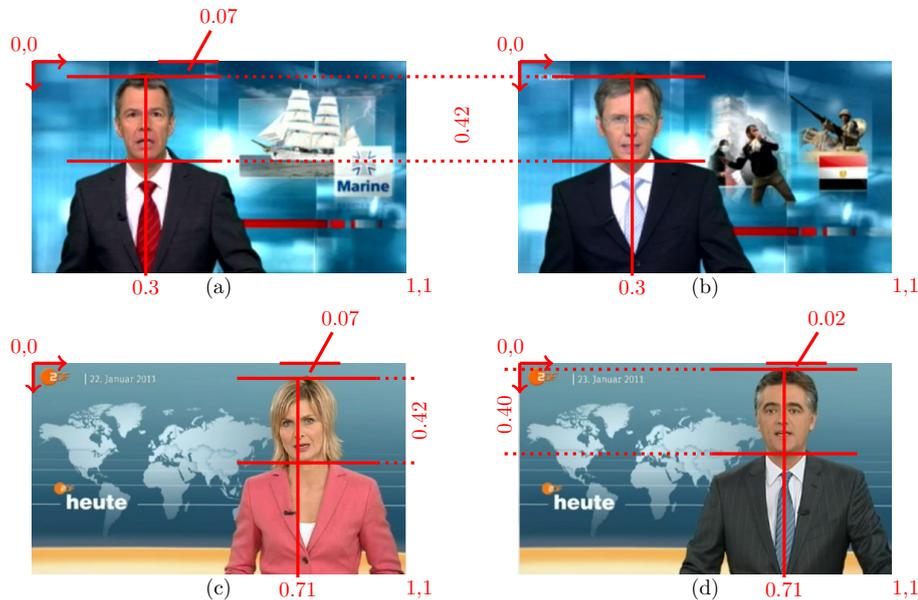


Abbildung 3.11: Vergleich der *Nahe-Einstellungen* in den Nachrichten von RTL (a) und (b) und ZDF (c) und (d). Die *tagesschau* verwendet keine *Nahe-Einstellungen*. Der Kopfraum beläuft sich zwischen 2% und 7% der Bildhöhe, die Position entspricht ungefähr der Drittel-Regel. Die Köpfe nehmen bei diesen Sendungen zwischen 40-42% der Bildhöhe ein.

schau abgebildet. Trotz der unterschiedlichen Moderatoren gleichen sich die Bilder bis ins Detail. Beide Moderatoren sind auf ca. 69% der Bildbreite positioniert und haben ca. 4% *Luft* zwischen dem Kopf des Moderators und der Oberkante des Bildes. Der Kopf der Moderatoren nimmt jeweils ca. 33% der Bildhöhe ein. Die Abbildungen für *RTL Aktuell* sind in Abbildung 3.12 (c) und (d) dargestellt. Beide Moderatoren befinden sich auf ca. 74% der Bildbreite und ihre Köpfe nehmen ca. 30% der Bildhöhe ein. Die *Luft* oberhalb der Köpfe ist in Bild (c) mit ca. 4% etwas größer als in Bild (d), indem die *Luft* bei ca. 3% der Bildhöhe liegt. In Abbildung 3.12 (e) und (f) sind die Einstellungen der *ZDF heute*-Nachrichten abgebildet. Auch hier befinden sich die Moderatoren für beide Bilder auf der gleichen horizontalen Position. Im Vergleich zu den anderen Nachrichtensendungen befinden sich die Positionen der Moderatoren hier mit 72% genau mittig zu den anderen. Die Kopfgröße liegt hier bei ca. 25-26% , die *Luft* über den Köpfen liegt bei ca. 5-6% der Bildhöhe.

Für Folgerungen aus einer *Amerikanischen-Einstellung* kann nur die Sendung *RTL Aktuell* verwendet werden, weil nur sie eine *Amerikanische-Einstellung* verwendet. Sie ist in Abbildung 3.13 (a) dargestellt. Der Moderator wird bei dieser Einstellung auf ca. 76% der Bildbreite positioniert. Er hat ca. 8% *Luft* über dem Kopf und dieser nimmt ca. 16% der Bildhöhe ein.

3.3 Vergleich der Nachrichten von ARD, ZDF und RTL aus der Sicht des Zuschauers



Abbildung 3.12: Vergleich der *Halbnahe-Einstellungen* der Nachrichtensendungen von ARD (a,b), RTL (c,d) und ZDF (e,f). Die Köpfe der Moderatoren nehmen zwischen 25-33% der Bildhöhe ein, die Luft über dem Kopf beträgt zwischen 3-6% der Bildhöhe. Die Moderatoren befinden sich in einem Bereich von ca. 69-74% der Bildbreite.

Die Messung der letzten zwei Werte wurden aufgrund der abwärts gerichteten Blickrichtung des Moderators relativ zur Mitte der gezeigten Kopfoberseite geschätzt.

Die Abbildungen 3.13 (b) und (c) zeigen die *Zweier-Einstellungen* der hier verglichenen Sendungen *RTL Aktuell* und *ZDF heute*. Die *tagesschau* zeigt keine *Zweier* und ist deshalb nicht aufgeführt. Bei *RTL Aktuell* befinden sich die Moderatoren auf ca. 30% und 76% der Bildbreite. Die Kopfgröße beläuft sich auf ca. 14-18%, ferner wurden zwischen ca. 12-16% *Luft* über den Köpfen des Moderators eingerichtet. Bei der *ZDF heute* befinden sich die Moderatoren auf ca. 30% und 79% der Bildbreite, die Luft über den Köpfen beträgt ca. 20-22% und die Kopfgrößen belaufen sich auf ca. 12-16%.

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

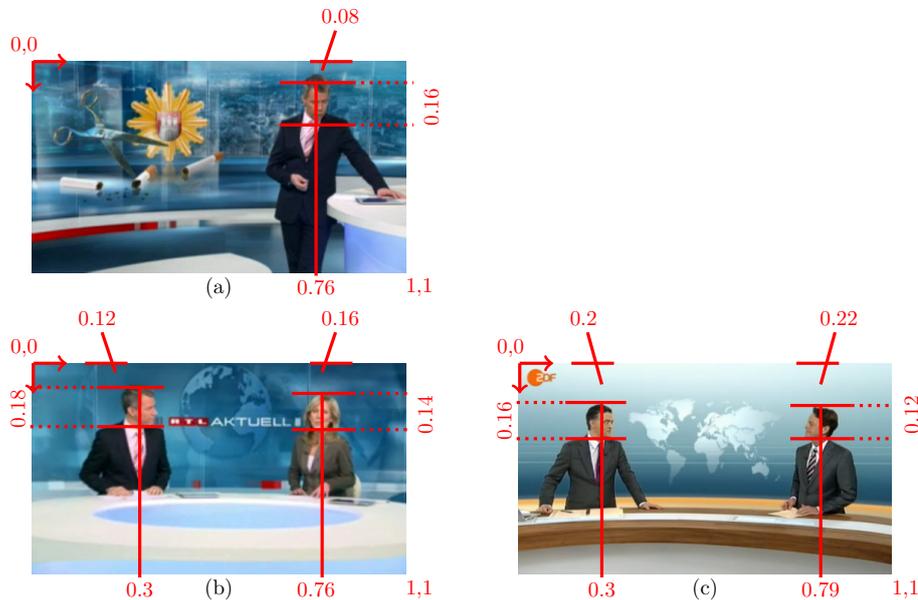


Abbildung 3.13: Vergleich der *Amerikanischen-* (a) und *Zweier-Einstellungen* von *RTL Aktuell* (b) und *ZDF heute* (c). Bei der *Amerikanischen-Einstellung* von *RTL* beträgt die *Luft* über dem Kopf ca. 8% und die Größe des Moderatorenkopfes ca. 16%. Für eine *Zweier-Einstellung* beträgt die *Luft* zwischen 12-22% der Bildhöhe. Die Köpfe nehmen ca. 12-18% der Bildhöhe ein.

Auf die Vorstellung der *Totale* der einzelnen Sender wird an dieser Stelle verzichtet. In dieser Arbeit halten wir uns an den Grundsatz, dass die *Totale* einen Überblick über das Studio gibt. Die Größe und Position der Moderatoren ist an dieser Stelle ein Resultat aus der Umgebung des Studios und wird als nicht aussagekräftig erachtet. Eine *Totale-Einstellung* zeigt gemäß Kapitel 3.1.2 alle relevanten Objekte im Bild.

3.3.3 Zusammenfassung der Auffälligkeiten der Einstellungsfolgen und der Einstellungsgrößen

Bei der Betrachtung der in Kapitel 3.3.1 dargestellten Graphen fällt auf, dass beginnend bei der *tagesschau*, über die *ZDF heute*-Sendung bis zu *RTL Aktuell*, die Anzahl der unterschiedlichen Einstellungen immer mehr zunimmt. Diese Tatsache kann vermutlich darauf zurückgeführt werden, dass die *tagesschau* die älteste der drei präsentierten Sendungsformen ist. Man kann davon ausgehen, dass aus technischen Gründen hier keine Kamerafahrten möglich sind. Das *ZDF* hat ein relativ neues Studio und damit ganz andere Optionen. Kamerafahrten sind hier möglich, werden aber aufgrund der eher konservativen Ausrichtung nicht übermäßig eingesetzt. Anders zu dieser Philosophie verhält es sich bei *RTL Aktuell*. Der Sender gehört zu den

3.3 Vergleich der Nachrichten von ARD, ZDF und RTL aus der Sicht des Zuschauers

privat-kommerziellen Sendern, hat das modernste Studio und versucht über einen modernen, „flippigen“ Auftritt das Unterhaltsame in den Vordergrund zu stellen. Viele Kamerapositionen und viel Bewegung dominieren hier das Erscheinungsbild.

Die einzelnen Sendungen verwenden typische Standard-Einstellungen. Diese sind fest definiert und werden auch sendungsübergreifend eingesetzt. Der Vergleich der Einstellungen zeigt, dass alle Sender ein gleiches Grundverständnis über die zu verwendenden Einstellungen besitzen. Für alle Sender ist die *Halbnahe* das Mittel der Wahl bei der Präsentation der Nachrichten. Diese wird, sofern technisch möglich, durch *Nahe-* oder *Amerikanische-* bzw. *Zweier-Einstellungen* ergänzt. Für die *Nahe-Einstellung* wird der Kopf des Moderators mit einer Größe von ca. 40-42% der Bildschirmhöhe gezeigt. Das Mittel liegt bei ca. 42%. Für eine *Halbnahe* liegen die Größen zwischen 25-33% (Mittel: ca. 30%). Die *Amerikanische-Einstellung* zeigt RTL mit einer Kopfhöhe von ca. 16% und die *Zweier-Einstellungen* werden bei den hier betrachteten Sendung mit einer Kopfgröße von ca. 12-18% (Mittel: ca. 15%) gezeigt. Für alle *Nahen-Einstellungen* ist die *Luft* zwischen 2% und 7% der Bildhöhe. Dieser Wert ändert sich auch für *Halbnahe-Einstellungen* nicht. Je größer der Öffnungswinkel der Kamera bei einer Einstellung ist, desto größer wird die *Luft* über dem Moderator. Sie vergrößert sich von ca. 8% für eine *Amerikanische-Einstellung* auf bis zu 22% für die *Zweier-Einstellung* bei der *ZDF heute*-Sendung.

Für die Positionen der Moderatoren lässt sich für die *Nahe* und *Halbnahe* eine klare Tendenz zu den Werten der Drittel-Regel erkennen. Für die Definition der einzelnen Einstellungen sollten also sowohl rechte und linke *Nahe-* bzw. *Halbnahe-Einstellungen* vorgesehen werden. Für die *Amerikanische-Einstellung* kann dieser Schluss aufgrund mangelnder Beispiele nicht automatisch gezogen werden. Die Position des Moderators lässt aber eine ähnliche Tendenz vermuten, sodass auch hier sowohl rechte wie auch linke Einstellungen vorgesehen werden sollten. Zusätzlich zu den jeweils seitlichen Ausprägungen der Einstellungen wird jeweils eine zentrale vorgesehen.

3. AUSGEWÄHLTE ASPEKTE BEI NACHRICHTEN- UND MAGAZINSENDUNGEN

Kapitel 4

Anforderungen an ein intelligentes automatisches Kamerasystem

Intelligentes Verhalten wird an den Möglichkeiten des Menschen gemessen. Die vorhergehenden Kapitel lassen erkennen, dass im Nachrichten- und Magazinbereich den menschlichen Kameramann insbesondere die Eigenschaften *Flexibilität* und die Fähigkeit *natürliche Sprache* zu verstehen auszeichnen. Für ein automatisches intelligentes Kamerasystem müssen die Aktionsmöglichkeiten eines Kamerasystems durch die zwei genannten Eigenschaften ergänzt werden. Im Folgenden werden die betriebsnotwendigen Aktionsmöglichkeiten sowie die Anforderungen hinsichtlich Sprachverständnis und Flexibilität vorgestellt.

4.1 Multiple Aktionen

Eine intelligente Steuerung sollte auf den für ein automatisches Kamerasystem möglichen Aktionen aufsetzen und sie um den Aspekt des menschenähnlichen Verhaltens erweitern. Ein Kamerasystem bietet, wie bereits in Kapitel 2.1 vorgestellt, die Möglichkeiten der Joystick-Interaktion, des Ladens und des Speicherns sowie die Möglichkeit Kamerafahrten anzustoßen. Die folgende Auflistung erklärt die Anforderungen, die hinsichtlich des Aktionspotentials an ein intelligentes Steuerungssystem gestellt werden:

1. *Joystick-Interaktion*: Mit Hilfe des Joysticks wird das Kamerasystem verfahren. In der *Einstellungsfindung* wird über den Joystick die am besten geeignetste Einstellung für

4. ANFORDERUNGEN AN EIN INTELLIGENTES AUTOMATISCHES KAMERASYSTEM

die jeweilige Sendung gesucht und gefunden. Es sollten mindestens die Achsen X, Y, Z, Pan, Tilt und Zoom unterstützt werden. Die Achse *Roll* ist nicht erforderlich, da im Nachrichten- und Magazinbereich kein *Roll* eingesetzt wird (vgl. Kapitel 3.1.3). Außerdem sollten Joystick-Aktionen solange durchführbar sein, bis bestimmte Bildeigenschaften erfüllt sind. In der *Sendekontrolle* erlauben die Joystick-Kommandos die Korrektur bestimmter Verfehlungen im Bild. Dabei müssen Quantitäten zum Steuern der Intensität der Ausführung ebenfalls berücksichtigt werden (vgl. Kapitel 3.2.1).

2. *Speichern/Laden*: Das Speichern ermöglicht es bestimmte Positionen für den späteren Gebrauch festzuhalten. Diese können sowohl für eine Diskussion der verschiedenen Positionen dienen, als auch für die Verwendung in einer Sendung gespeichert werden. Ein Kameramann benutzt beim Einstellen einer bestimmten Position die zuvor festgelegte Bodenmarkierung als Ausgangspunkt und stellt dann das Bild für die individuellen Studioverhältnisse bestmöglich ein. Ein intelligent gesteuertes Kamerasystem muss in der Lage sein Positions- und Bildinformationen zu speichern, um bei einem Wiederanfahren der Position reproduzierbare Ergebnisse zu erzielen. Neben zuvor gespeicherten Individual-Einstellungen sollen ebenfalls wiederkehrende Standard-Einstellungen (vgl. Kapitel 3.1.2) unterstützt werden. Für die Einstellungen *Nahe*, *Halbnahe* und *Totale* muss eine rechte, mittlere und eine linke Version dem System bekannt und vermittelbar sein.
3. *Zeitwährende Aktionen*: Joystick-Aktionen sind Aktionen, die in der Regel nach Erfüllung beendet werden können. Die Kamera soll bei Joystick-Aktionen nach Ausführung einer bestimmten Bewegung still stehen. Die Aktion selbst kann danach aus dem System entfernt werden. Andere Aktionen besitzen einen zeitlichen Bezug und müssen solange im System bleiben, bis die Ausführung explizit beendet wird. Dies ist immer dann der Fall, wenn über einen längeren Zeitraum eine bestimmte Einstellung gehalten werden muss. Eine intelligente Kamerasteuerung sollte in der Lage sein, eine bestimmte Bildkomposition über einen längeren Zeitraum aufrecht zu erhalten. In Kapitel 3.2.2 ist beispielhaft dargestellt, wie Kamera1 über den gesamten Sendungszeitraum eine bestimmte Einstellung aufrecht erhält.
4. *Bildbasierte Fahrten*: Fahrten müssen bildbasiert ausgeführt werden. Sie verlaufen im Nachrichtenbereich immer von einer Einstellung auf eine andere und müssen dabei einen Fixpunkt berücksichtigen. Als Fixpunkt sollte mindestens die *Luft* über dem Kopf vom System erfasst werden. Die von einem Kamerasystem „mitgebrachte“, vorinterpolierende

Funktion *Fahrt* ist für eine intelligente Steuerung nicht verwendbar. Bei der Interpolation der Stützstellen werden keine Bildinformationen berücksichtigt. Fahrten sollten deshalb als Kombination von Joystick-Interface und Bildverarbeitung ausgeführt werden um so die Bildeigenschaften berücksichtigen zu können.

Der Mensch beherrscht oben genannte Aktionen bereits durch Training und Ausbildung. Für eine automatische Steuerung müssen diese Anforderungen jedoch explizit gefordert werden.

4.2 Natürlichsprachliche Kommandos

Für ein menschenähnliches Verhalten einer intelligenten Kamera steht die verbale Instruierbarkeit im Vordergrund. Dies ist die natürliche Form der Kommunikation. Wie bei menschengeführten Kameras soll die Regie mit dem System kommunizieren und dabei verbal Anweisungen an dieses richten können. Die Projektion der gewünschten Änderung auf die jeweiligen Geräteachsen übernimmt die intelligente Kamerasteuerung. Herkömmliche Kamerasysteme bieten unterschiedliche Achsen zum Verfahren der Kamera an. Der Regisseur muss hier von den gewünschten Bildänderungen auf die jeweiligen Achsen schließen, die für eine gewünschte Änderung im Bild zuständig sind. Der notwendige Rückschluss sollte von einer intelligenten Steuerung automatisch erfolgen. Die Bedeutung des Bildes muss wieder in den Vordergrund treten. Da die natürliche Sprache eine Vielzahl von Möglichkeiten bietet Anweisungen zu generieren, muss eine abgeschlossene Menge von Kommandos definiert werden, die typische Situationen der Kameraarbeit im Studio abdeckt. Zu diesen zählen Kommandos zum Bewegen und Stoppen des Kamerasystems sowie zum Anfahren und Speichern bestimmter Einstellungen und Positionen:

1. *Stopp-Kommandos* erlauben es, das Kamerasystem an jeder beliebigen Stelle zu stoppen. So können Fahr-Anweisungen bereits vorzeitig, bei Erreichen der gewünschten Position, abgebrochen werden.
2. *Joystick-Kommandos* dienen dem Steuern des Kamerasystems. Das Kamerasystem wird in der Phase der *Einstellungsfindung* verbal durch die Regie dirigiert, bis eine Einstellung gefunden ist, die den technischen und ästhetischen Anforderungen genügt. Ergänzend dazu geben die Joystick-Kommandos dem Regisseur die Möglichkeit, die gebotene Studiowelt durch das Auge der Kamera zu sehen und so die Wirkung des Studios auf den Zuschauer zu erproben. Aus diesem Grund müssen alle klassischen Bewegungsachsen der Kamera verbal

4. ANFORDERUNGEN AN EIN INTELLIGENTES AUTOMATISCHES KAMERASYSTEM

abgebildet werden. Zusätzlich zu diesen Kommandos müssen Anweisungen berücksichtigt werden, die sich aus dem Bildeindruck des Regisseurs ergeben. Durch die Separation von Regie und Studio steht dem Regisseur nur der aufgenommene Bildausschnitt der Kamera als Entscheidungsgrundlage zur Verfügung. Aus diesem Grund müssen neben den klassischen Bewegungsachsen auch Joystick-Kommandos anhand von Bildänderungen beschrieben werden können.

Die Kommandos sollten quantitative Aussagen unterstützen. Unbestimmte Quantitäten variieren dabei die Fahrtlänge. Zu unterscheiden sind mindestens *viel*, *normal* und *wenig*. *Kamera1 fahr ein bisschen näher ran* aus Kapitel 3.2.1 ist ein Beispiel für ein derartiges Joystick-Kommando. Zusätzlich müssen bestimmte Quantitäten unterstützt werden, die, eine bestimmte Einheit vorausgesetzt, das Kamerasystem um die genannte Distanz bewegen.

3. *Objekt-bezogene Kommandos* erlauben es, die Erscheinung einzelner Objekte im Bild zu ändern. Typischerweise fallen die horizontale Position sowie die *Luft* um ein Objekt in diese Kategorie. Die horizontale Position ist dabei quantitativ und qualitativ anzugeben. Die *Luft* kann allgemein auf ein Objekt bezogen oder speziell, in Bezug auf eine Richtung zum Objekt hin, angegeben werden.
4. *Konditional Kommandos* veranlassen eine Aktion, bis ein bestimmtes Ereignis eintritt. *Kamera1 gehe etwas höher, bis die Monitorecke aus dem Bild ist*, wäre ein Beispiel dafür.
5. *Speichern Kommandos*: Das Speichern von Einstellungen hat das Ziel, diese während der Sendung immer wieder anfahren zu können. *Speichern Kommandos* ersetzen die konventionellen *Bodenmarkierungen* im Studio. Das System fährt auf eine bestimmte Position und richtet die Einstellung so ein, dass sie mit der ursprünglich ausgewählten Einstellung bestmöglich übereinstimmt. Dies sollte auch für geänderte Studiobedingungen möglich sein.
6. *Kommandos zum Anfahren von Positionen und Einstellungen*: Einmal gespeicherte Positionen sollten immer wieder verbal angefahren werden können. Das gilt sowohl für zuvor gespeicherte Positionen als auch für Standard-Einstellungen, wie in Kapitel 3.1.2 dargestellt.
7. *Zeitwährende Kommandos* erlauben es, bestimmte Einstellungen über einen längeren Zeitraum zu erhalten. Diese Option ist immer dann interessant, wenn sich der Moderator durch

das Studio bewegt. Es entsteht eine Interaktion zwischen Moderator und Kamera, bei der die Kamera dem Moderator folgt.

All diese Kommando-Typen sollten in der Form [Adressat] [Kommando] gegeben werden. So wird sichergestellt, dass das System auch in Studios mit mehreren Kameras sicher eingesetzt werden kann.

Die verbale Steuerbarkeit ist der Grundpfeiler eines intelligenten Kamerasystems. Erst die direkte Kommunikation mit einer Maschine kann diese zu einer intelligenten Maschine machen. Der Schwerpunkt der vorliegenden Arbeit liegt deshalb auf einer intelligenten Lösung zur verbalen Kommunikation zwischen Kamera und Regie.

4.3 Hohe Flexibilität

Flexibilität wird im Kontext dieser Arbeit in zweierlei Hinsicht verstanden. Zum einen bedeutet Flexibilität die Variabilität der Vorgaben und zum anderen, dass das System individuell auf Änderungen in der Umwelt reagieren kann.

4.3.1 Variabilität der Arbeitsbegriffe

Für die Integration einer intelligenten Kamerasteuerung in eine Studioumgebung müssen dem System typische Begriffe aus der Fernsehwelt bekannt sein. Allen voran sind hier die Standard-Einstellungen zu nennen. Das fünf-stufige Modell aus Kapitel 3.1.2 stellt hier einen zweckmäßigen Ausgangspunkt dar. Ausschlaggebend ist jedoch, dass alle Einstellungen, die für eine Sendung geplant wurden, dem System bekannt sind. Außerdem muss es möglich sein, die einzelnen Einstellungen individuell anzupassen. Jeder Sender hat seine eigenen Vorstellungen, wie z.B. eine *Halbnahe-Einstellung* auszusehen hat. Um das System universell einsetzen zu können, muss hier die Möglichkeit bestehen, eigene Modifikationen zu einer bestimmten Position in das System einzubringen. Neben den Eigenschaften der Standard-Einstellungen muss das System typische Elemente der Einstellungseinrichtung, wie z.B. Safe-Title und Safe-Action kennen. Die Möglichkeit des Ausrichtens von Studioobjekten an diesen Elementen erleichtert die Arbeit mit dem Kamerasystem. Zu den typischen Eigenschaften einer Einstellung gehört es, die Augenhöhe des Moderators zu berücksichtigen. Das System sollte diese kennen, um so eine Einstellung automatisch auf die Augenhöhe des Moderators vornehmen zu können.

4. ANFORDERUNGEN AN EIN INTELLIGENTES AUTOMATISCHES KAMERASYSTEM

4.3.2 Kompensationsstrategie

Bisherige automatische Kamerasysteme speichern Einstellungen anhand der Achsdaten, die das System beim Festlegen der Position angefahren hatte. Diese Form der Speicherung bietet zwar die Chance einen immer gleichen Seh-Eindruck zu generieren, aber gleichzeitig darf sich die Umwelt nicht ändern (vgl. Kapitel 1.2). Unterschiedlich große Moderatoren können mit einer einzigen Einstellung ebenso wenig abgebildet werden, wie unvorhergesehene Änderungen an der Studio-Umgebung.

Bei manueller Kameraführung werden Studiomarkierungen verwendet, um die Ausgangsposition einer Einstellung zu definieren. Kleinere Änderungen nimmt der Kameramann selbst vor. Solche kleinen Änderungen können von automatischen Systemen (bisher) nicht selbstständig kompensiert werden. Ein intelligentes System muss die quantitativen Bildeigenschaften speichern und auf potentielle Verfehlungen im Bild reagieren können.

Die dafür notwendige Kompensationstrategie muss die Position auf die aktuellen Umstände im Studio – z.B. an eine andere Moderatorengröße – selbstständig anpassen. Die Achsen Pan, Tilt und Zoom können hierfür verwendet werden, weil diese Achsen nicht die Perspektive der gewünschten Einstellung ändern.

Kapitel 5

Automatische Verarbeitung von natürlicher Sprache im Studio

Teamorientierte Arbeitsumgebungen sind auf Kommunikation angewiesen. Die natürliche Sprache ist für den Menschen eine effiziente und universell einsetzbare Form der Kommunikation. Zum einen können beliebige Aspekte der Kameraführung formuliert werden und zum anderen können sich beteiligte Akteure, durch Verwendung natürlicher Sprache, stets auf ihre primäre Aufgabe innerhalb des Teams konzentrieren. Im Fernsehstudio bedeutet das auf der einen Seite, dass der Regisseur, während er Anweisungen gibt, stets auch auf andere Dinge, wie z.B. den Gesamteindruck der Sendung, achten kann und auf der anderen Seite, dass es für den Kameramann möglich ist, während er eine bestimmte Einstellung zeigt, Anweisungen aus der Regie entgegenzunehmen. Für ein intelligentes Steuerungssystem wird es deshalb in der vorliegenden Arbeit als essentiell angesehen, dass sich das System nahtlos in eine solche verbal-dominierte Umgebung einfügen kann. Die folgenden Kapitel zeigen, wie mit Hilfe einer kombinatorischen Kategorial-Grammatik natürliche Sprache in eine logische Form transformiert werden kann und wie mit deren Hilfe das Kamerasystem verbal gesteuert wird.

Um den Moderator nicht zu irritieren, ist das Studiopersonal gehalten, während einer Sendung nicht zu sprechen. Somit erfolgt die Kommunikation zwischen Regie und Kameramann einseitig von der Regie aus. Alle Anweisungen an den Kameramann beziehen sich auf das aktuell von ihm gezeigte Bild. Dieses wird in der Regie auf Kontrollmonitoren angezeigt. Ein Abfragen des jeweils aktuellen Status der Kamera ist nicht nötig. Die Bildgenerierung im Studio basiert auf der Teamarbeit zwischen Regie und Kameramann. Es bedarf nicht vieler Worte um eine Einstellung so anzupassen, dass sie sich in das Gesamtkonzept einer Sendung einfügt. Dafür

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

haben beide Kommunikationspartner den gleichen Wissensstand. Eine intelligente Steuerung wird hier deshalb als reiner Kommando-Empfänger verstanden. Ein Dialog-System ist nicht erforderlich und würde die Abläufe im Studio bremsen.

Die Kommunikation zwischen Regie und Kamera erfolgt im Imperativ durch weitestgehend einheitliche Kommandos. Für einen höflichen Austausch bleibt im Live-Betrieb, wie er hier angenommen wird, keine Zeit. In Studios und Sendungen mit mehreren beteiligten Kamerasystemen hat sich eine bestimmte Struktur der Kommandos etabliert. Jedem Kommando wird der Rezipient des Kommandos vorangestellt (vgl. Kapitel 3.2.3). So können über einen einheitlichen Kanal (z.B. Studiofunk) unterschiedliche Kamerasysteme angesprochen werden. Wir führen diese Struktur als verbindlich für das hier vorgestellte System ein. Abbildung 5.1 zeigt diese Struktur schematisch.

[ACTOR] [CMD]

Abbildung 5.1: Basis Struktur eines Studio-Kommandos. Dem Kommando [CMD] wird der Rezipient [ACTOR] vorangestellt.

Das verwendete Schema lehnt sich an die Backus-Naur-Form (BNF) [Knu64] an. Terminalwörter sind dabei klein im Infinitiv angegeben. Nicht-Terminalwörter werden in Großbuchstaben dargestellt. In der Darstellung steht ACTOR für den Rezipienten des Kommandos (z.B. „Kamera1“) und CMD für die Anweisung selbst.

Anhand von Interviews (vgl. Kapitel 3.2.3) und durchgeführten Testläufen wurden unterschiedliche Typen von Anweisungen, die zur Steuerung einer Kamera nötig sind, identifiziert. Der Fokus der im Folgenden beschriebenen Anweisungen liegt auf der Einstellungsfindung sowie auf der Verwendbarkeit der Kommandos für Nachrichten- und einfache Magazin-Produktionen. Sie zeichnen sich durch (relativ) einfache Studio-Designs und starr definierte, stehende Einstellungen aus, die – einmal festgelegt – immer wieder angefahren werden. Im Rahmen der Einstellungsfindung liegt der Fokus vor allem auf der Steuerbarkeit des Kamerasystems unter Verwendung von studiotypischen Anweisungen. Hierzu zählen Joystick-Kommandos zum Verfahren des Kamerasystems, Kommandos zum Platzieren von Objekten im Bild sowie Konditional-Anweisungen, die solange ausgeführt werden, bis eine bestimmte Bedingung erfüllt ist. Zur besseren Handhabbarkeit gibt es zusätzlich noch Speicher-Kommandos, mit denen einmal gefundene Kamerapositionen gespeichert werden können sowie Lade-Anweisungen, die gespeicherte Einstellungen wieder herstellen.

Die Grammatik beschreibt grundlegende Kommandos in natürlicher Sprache, wie sie bei der täglichen Arbeit in Studioumgebungen zur Instruktion von Kameraleuten verwendet werden. Verben werden hier im Infinitiv referenziert. In der Grammatik selbst werden sie jedoch im Indikativ verwendet. Auf eine vollständige Deklination der einzelnen Verben wurde in dieser Arbeit verzichtet. Im Folgenden werden Wörter der natürlichen Sprache *kursiv* dargestellt. Bezeichner für Kategorien stehen in *Courier*.

Jedes Wort der Grammatik wird einer bestimmten Kategorie zugeordnet. Diese Kategorie kann mehrere Ableitungsmöglichkeiten umfassen. Im Text wird deshalb sowohl von der Zuordnung eines Wortes zu einer bestimmten Kategorie gesprochen als auch vom Einfügen einer bestimmten Ableitungsvorschrift für diese Kategorie.

In den folgenden Kapiteln werden die einzelnen Kommandos beginnend mit dem Stopp-Kommando vorgestellt. Das Stopp-Kommando (Kapitel 5.1) zählt nach Ansicht des Autors zu den wichtigsten und gleichzeitig einfachsten Kommandos und führt deshalb die Aufzählung an. Es schließen sich die primär für die *Einstellungsfindung* nötigen Kommandos an. Beginnend mit den verschiedenen Möglichkeiten für Joystick-Kommandos zur direkten Steuerung der Kamera, folgen objektbezogene Kommandos, die Bewegungen anhand eines bestimmten Objekts einfordern sowie Konditionalkommandos, die Bewegungen beschreiben, die solange ausgeführt werden sollen, bis eine bestimmte Bild-Eigenschaft¹ oder Welt-Eigenschaft² erfüllt ist. Das Speichern-Kommando schließt die Menge der für die *Einstellungsfindung* verwendbaren Kommandos. Die danach vorgestellten Kommandos zur Anfahrt einer Standard-Einstellung oder einer zuvor gespeicherten Einstellung sowie zeitwährende Kommandos, zur Einhaltung einer bestimmten Bildkomposition über einen längeren Zeitraum, werden nach Meinung des Autors primär in der Phase der *Sendekontrolle* verwendet. Sie sind deshalb am Ende des Kapitels aufgeführt. Die einzelnen Kapitel folgen im Wesentlichen der gleichen Struktur. Ausgehend von beliebig ausgesuchten Beispielen, wird der Aufbau des Kommandos in oben bereits vorgestellter Struktur (ähnlich BNF) angegeben. Daran schließt sich eine Ableitung in CCG für ein typisches Beispiel des Kommandos an. Die Kapitel schließen jeweils mit den semantischen Ergänzungen sowie der Darstellung des, von der verwendeten Grammatik generierten, logischen Ausdrucks. Der logische Ausdruck wurde dabei durch die im Prototypen verwendete Bibliothek OpenCCG erzeugt und in die Arbeit aufgenommen.

¹Eine Bildeigenschaft ist z.B. eine bestimmte Position im Bild für ein bestimmtes Objekt.

²Eine Welteigenschaft wäre z.B. eine bestimmte Höhe für die Kamera.

5.1 Stopp-Kommando

Mit Hilfe des Stopp-Kommandos hält die Kamera an und beendet alle evtl. ausstehenden Fahr-befehle. In Gefahren-Situationen, wenn also die Kamera nicht wie erwartet reagiert oder falsch angewiesen wurde, ist dieses Kommando ebenso wichtig, wie bei der Feinjustage, wenn ein gewünschtes Bild erreicht wurde, bevor das System von sich aus stoppt. Ein Stopp-Kommando wird sowohl durch den direkten Wortgebrauch als auch über Synonyme, wie *Danke*, *Halt* oder *Gut so* initiiert. Die Kommandos werden als Interjektion (lat. für Einwurf) bezeichnet, da sie nicht aus vollständigen Sätzen bestehen. *Stopp* ist das einzige Kommando, bei dem von der Struktur in Abbildung 5.1 abgewichen werden darf. Die Interjektion darf auch ohne vorhergehenden ACTOR verwendet werden. Abbildung 5.2 zeigt die Satzstruktur des Stopp-Kommandos.

$$\begin{aligned} [\text{STOP-CMD}] &:= [\text{INTERJECTION}] \\ [\text{STOP-CMD}] &:= [\text{ACTOR}] [\text{INTERJECTION}] \end{aligned}$$

Abbildung 5.2: Struktur eines Stopp-Kommandos.

Sie besteht entweder aus einem Wort, der Interjektion selbst oder aus der Kombination von Akteur und Interjektion. Fehlt der Akteur vor der Interjektion, gilt das Stopp Kommando für alle beteiligten Kamerasysteme. Ist er explizit erwähnt, gilt das Kommando nur für den referenzierten Akteur. Abbildung 5.3 zeigt die Ableitung des Beispiel-Satzes *Kamera1 Stopp* in CCG.

$$\frac{\frac{Kamera1}{ACTOR} \quad \frac{Stopp}{S \backslash ACTOR}}{S} <$$

Abbildung 5.3: Ableitung des Stopp-Kommandos durch CCG.

Der Sinn dieses Satzes erschließt sich ausgehend von der Interjektion. Die Kategorie der Interjektion (hier: *Stopp*) wird deshalb durch $S \backslash ACTOR$ ausgedrückt. Ein Satz kann damit genau dann abgeleitet werden, wenn der Interjektion selbst ein ACTOR vorausgeht. Für die Ableitbarkeit ohne vorausgehenden Akteur wird der Kategorie der Interjektion zusätzlich die Vorschrift S hinzugefügt. So kann die Interjektion ebenfalls für sich allein abgeleitet werden. *Kamera1* wird der atomaren Kategorie ACTOR zugeordnet.

Für die logische Ableitung des Ausdrucks wird die Kategorie ACTOR durch $\models @_{actor}(X)$ ergänzt und die Kategorie von *Stopp* mit $\models @_{action}(stop \wedge INTERJECTION_Y \wedge ACTOR_X)$

erweitert. Diese Ausdrücke entsprechen den bereits in Abbildung 2.13 vorgestellten Formulierungen in CCG. Im Weiteren wird nur noch die hybrid-logische Form angegeben. Der sich ergebende Ausdruck beschreibt eine Stopp-Aktion, die neben dem Adressaten auch die Interjektion selbst in Form von Nominalen enthält. Die jeweils entscheidenden Nominalen sind im Weiteren in Großbuchstaben dargestellt. Es ergibt sich für das Beispiel die in Abbildung 5.4 dargestellte logische Form.

$$\begin{aligned} @w1:action(stop \wedge \\ & \langle INTERJECTION \rangle Stopp \wedge \\ & \langle ACTOR \rangle (w0:actor \wedge Kamera1 \wedge \\ & \quad \langle num \rangle sg-X \wedge \\ & \quad \langle gender \rangle genderFem-X)) \end{aligned}$$

Abbildung 5.4: Logische Form des Stopp-Kommandos.

ACTOR wird durch die Attribute `num` (Numerus) und `gender` ergänzt. Wird kein ACTOR angegeben, so gilt das Kommando für alle an der Produktion beteiligten Kamerasysteme. Für die Behandlung dieses Falls gibt es zwei Möglichkeiten: Der logische Ausdruck wird vom System um einen vordefinierten Spezial-Terminus *allCams* ergänzt. Die logische Form gestaltet sich dann zu $\models @_{action}(stop \wedge INTERJECTION_Y \wedge ACTOR_{allCams})$. Alternativ kann das Nominal *ACTOR* in der logischen Form weggelassen werden. Die Kategorie der Interjektion wird dann durch $\models @_{action}(stop \wedge INTERJECTION_Y)$ ergänzt. Die zuletzt genannte Version wird hier präferiert. Dabei wird die Auffassung vertreten, dass bei der Verarbeitung natürlicher Sprache keine nicht-geäußerten Elemente automatisch hinzugefügt werden sollten. So kann in späteren Verarbeitungsschritten immer auf den tatsächlich geäußerten Satz geschlossen werden.

5.2 Joystick-Kommandos

Obwohl man vermuten könnte, dass eine intelligente Kamerasteuerung ohne Joystick-Kommandos auskommt, haben die im Rahmen dieser Arbeit durchgeführten Interviews deutlich gemacht, dass diese Kommandos bei der täglichen Arbeit an der Kamera sehr häufig zum Einsatz kommen. Sie werden sowohl für kleine Korrekturen kurz vor der Sendung, als auch bei der Einstellungsfindung verwendet. Das folgende Kapitel weicht von der in Kapitel 5 vorgestellten Kapitelstruktur ab. Aufgrund der umfangreichen Möglichkeiten und der sich daraus ergebenden Komplexität werden in diesem Kapitel zuerst die jeweiligen Formen der Ansprache

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

für die verschiedenen Achsen eines Kamerasystems erklärt. Danach werden die unterschiedlichen Möglichkeiten der Richtungsbeschreibung in gleicher Weise, wie in den anderen Kapiteln vorgestellt. Das Kapitel schließt mit der Behandlung von bestimmten und unbestimmten Quantitäten.

5.2.1 Bewegungsachsen

Ein Kamerasystem folgt bei Bewegungen imaginären Achsen. Diese Achsen stehen für positionsändernde (X, Y und Z), orientierungsändernde (Pan, Tilt) und linsensteuernde (Zoom) Bewegungen. Die verbalen Beschreibungen dieser Bewegungen werden in diesem Kapitel erläutert.

Die Richtungen *vor*, *nach hinten*, *rechts*, *links*, *hoch* und *runter* gehören zu den positionsändernden Bewegungen eines Kamerasystems. Anweisungen zur Umpositionierung der Kamera werden mit Hilfe der Bewegungsverbren *fahren* oder *gehen* eingefordert. Abbildung 5.5 zeigt Beispiel-Kommandos für die Bewegung entlang dieser Achsen. Die Achsen *vor*, *nach hinten*, *rechts* und *links* werden anhand der Orientierung der Kamera definiert. *vor* und *nach hinten* beschreiben Bewegungen entlang der Kameralängsachse (in das Bild hinein, bzw. aus dem Bild heraus). *Rechts* bzw. *links* bezeichnet Bewegungen, die horizontal orthogonal dazu verlaufen. Die Achsen *hoch* und *runter* verlaufen unabhängig von der Orientierung der Kamera immer vertikal.

Kamera1 fahre hoch
Kamera1 gehe tiefer

Abbildung 5.5: Beispiele für positionsändernde Joystick-Kommandos.

Die Verwendung des Begriffs *zurück* kann missverständlich sein. Dieses Wort kann sowohl als invertierte letzte Aktion als auch als Bewegung vom Zielpunkt weg verstanden werden. Dieses Problem wird hier durch die Verwendung von Synonymen umgangen. Im vorliegenden System wird der Begriff *zurück* ausschließlich für die Invertierung der letzten Aktion verwendet. Die Phrasen *weiter weg* sowie *näher ran* beschreiben Bewegungen entlang der Kameralängsachse (siehe Kapitel 5.2.3).

Die Achsen Pan und Tilt sind orientierungsändernde Achsen und wirken sich auf die horizontale (Pan) bzw. vertikale (Tilt) Ausrichtung des Kamerakopfes aus. Soll die Orientierung des Kopfes geändert werden, so geschieht dies für beide Achsen über das Verb *schwenken*. Abbildung 5.6 zeigt Beispiele für solche Joystick-Kommandos.

Kamera1 schwenke hoch
Kamera1 schwenke nach rechts

Abbildung 5.6: Beispiele für orientierungsändernde Joystick-Kommandos.

In der umgangssprachlichen Verwendung werden auch die Begriffe *pane* bzw. *tilte* verwendet. Die Anweisung lautet dann *Kamera1 pane/tilte nach oben*. Nachdem diese Verben jedoch in keinem Duden zu finden sind, werden sie hier nicht weiter verwendet, obwohl die Satzstruktur bei der Verwendung der umgangssprachlichen Form gleich bleibt. Diese Verben könnten deshalb prinzipiell hinzugefügt werden. Mit Hilfe der linsensteuernden Achsen können Brennweite (Fokus) und Öffnungswinkel (Zoom) verändert werden. Die Brennweite ist bei großen Öffnungswinkeln kaum zu erkennen und wird deshalb selten angesteuert. Es gibt für das Verfahren der Brennweite kein eigenes Kommando. Der Fokus wird am Ende der Einstellungsfindung durch einen Satz wie: *Kamera1 stell das mal scharf* oder *Kamera1 zieh mal den Fokus gerade* angesprochen. Kommandos, die sich auf die Brennweite beziehen, werden hier nicht weiter verfolgt. Es wird davon ausgegangen, dass die Brennweite automatisch richtig justiert ist. Anders verhält es sich für die Ansteuerung des Zooms. Abbildung 5.7 zeigt Beispiele, wie der Zoom verbal angesteuert werden kann.

Kamera1 fahre offener
Kamera1 fahre enger

Abbildung 5.7: Beispiele für linsensteuernde Joystick-Kommandos.

Das Adverb *offener* verlangt eine weitwinkligeren, *enger* dagegen eine weniger weitwinkligeren Einstellung. Neben den hier vorgestellten Kommandos gibt es noch eine umgangssprachliche Variante, die die jeweilige Achse direkt benennt. Das Kommando lautet dann *Kamera1 zoome etwas raus/rein*. Da *zoome* jedoch nicht als deutsches Verb geführt wird, findet es hier keine Verwendung.

Die oben genannten Bewegungsachsen und deren verbale Verwendung sind die Basis für die Gruppe der Joystick-Kommandos. Aus den vorgestellten Beispielen lässt sich bereits ableiten, dass ein Joystick-Kommando immer ein *Bewegungsverb* enthält, von dem ausgehend mit einer zusätzlichen Richtungsangabe auf die jeweilige Achse geschlossen werden kann. Formen von *Schwenken* modifizieren dabei die Orientierung, Formen von *Fahren* kennzeichnen Änderungen an der Linse bzw. an der Position der Kamera. Dabei kann anhand der Richtung unterschieden

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

werden, ob die Linse oder die Kamera verfahren werden soll. Formen von *Gehen* werden nur für Positionsänderungen verwendet. Joystick-Kommandos werden oft durch die Angabe der Quantität der gewünschten Bewegung ergänzt. Näheres dazu in den Kapiteln 5.2.5 bzw. 5.2.6. Die in den Abbildungen 5.5 bis 5.7 aufgeführten Beispiele zeigen bereits, dass ein Joystick-Kommando immer in der gleichen Form aufgebaut ist. Abbildung 5.8 zeigt diese Struktur.

[JOYSTICK-CMD] := [ACTOR] [fahren|gehen|schwenken] [DIRECTION]

Abbildung 5.8: Struktur der Joystick-Kommandos. Den verschiedenen Verben geht der Akteur ACTOR voraus. Die gewünschte Richtung DIRECTION folgt ihnen.

Dabei bestimmt das Verb die Aussage des Satzes. Ihm geht ein Rezipient ACTOR voraus. Die Richtung der Bewegung DIRECTION folgt dem Verb. Tabelle 5.1 zeigt den Zusammenhang der oben erklärten Richtungen und Achsen.

Tabelle 5.1: Zuordnung von Verben auf Bewegungsachsen. Der Index o gibt an, dass diese Achse von der Orientierung der Kamera abhängt.

	fahren	gehen	schwenken
hoch / nach oben	Z	Z	TILT
runter / nach unten	$-Z$	$-Z$	-TILT
vor / nach vorne / näher ran	X_o	X_o	-
nach hinten / weiter weg	$-X_o$	$-X_o$	-
zurück	-	-	-
links / nach links	Y_o	Y_o	PAN
rechts / nach rechts	$-Y_o$	$-Y_o$	-PAN

Die Achse X_o verläuft horizontal entlang der Kameralängsachse. Der Index o zeigt an, dass die Achse von der Orientierung der Kamera abhängt. Die Achse Y_o verläuft ebenfalls abhängig von der Orientierung der Kamera horizontal orthogonal zu X_o . Die Achse Z verläuft vertikal. *zurück* wird als Spezialfall behandelt und beschreibt in diesem System keine Verfahr-Achse, sondern die Invertierung des letzten Kommandos. Die Richtungsbegriffe *nach hinten* bzw. *weiter weg* übernehmen die Funktion und Bedeutung der Richtungsangabe *zurück*.

5.2.2 Richtung durch richtungsbeschreibende Adverben

Eine Bewegungsrichtung kann durch ein Adverb, das eine bestimmte Richtung beschreibt, angegeben werden. Dies ist z.B. für *vor*, das umgangssprachliche *runter*, *hoch* und einige andere

mehr der Fall. Abbildung 5.9 zeigt die Ableitung für diese Adverben am Beispiel von *Kamera1 fahre hinauf*.

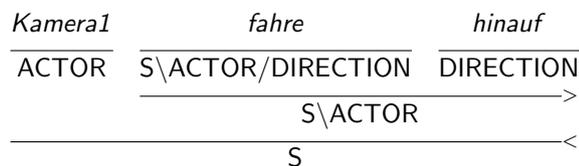


Abbildung 5.9: Ableitung eines richtungsbeschreibenden Adverbs.

Die Kategorie von *fahre* (S\ACTOR/DIRECTION) legt fest, dass für eine zulässige Ableitung dem Verb ein ACTOR vor- und eine DIRECTION nachsteht. Richtungsbeschreibende Adverben können jetzt direkt der Kategorie DIRECTION zugeordnet werden. Für die logische Form kann das jeweilige Wort direkt übernommen werden. Die Kategorie für die richtungsbeschreibenden Adverben wird deshalb um $\models @ (X)$ erweitert. Dem Satisfaction-Operator @ wird kein näher spezifizierender Index zugeordnet. Das Adverb steht für eine Richtung und kann so in die logische Form übernommen werden. Die Kategorie *fahre* wird in Zusammenhang mit richtungsbeschreibenden Adverben um folgende Vorschrift ergänzt: $\models @_{action}(joystick \wedge VERB_F \wedge DIRECTION_Y \wedge ACTOR_X)$. So ergibt sich die in 5.10 dargestellte logische Form für obige Anweisung.

```

@w1:action(joystick ^
  <VERB>fahre ^
  <ACTOR>(w0:actor ^ Kamera1 ^
    <num>sg-X ^
    <gender>genderFem-X) ^
  <DIRECTION>(w2 ^ hinauf))

```

Abbildung 5.10: Logische Form von Joystick-Kommandos, deren Richtung durch ein richtungsbeschreibendes Adverb bestimmt wird.

Diese beschreibt eine Joystick-Aktion durch die Nominale VERB, ACTOR und DIRECTION. Die Attribute **num** und **gender** haben in der dargestellten Ableitung keine Bedeutung und sind nur der Vollständigkeit halber abgebildet.

Form wird es hier ebenfalls als Richtung verstanden und abgeleitet. Sie wird dann allerdings vom Handlungsmodell unterschiedlich interpretiert (s. Kapitel 6).

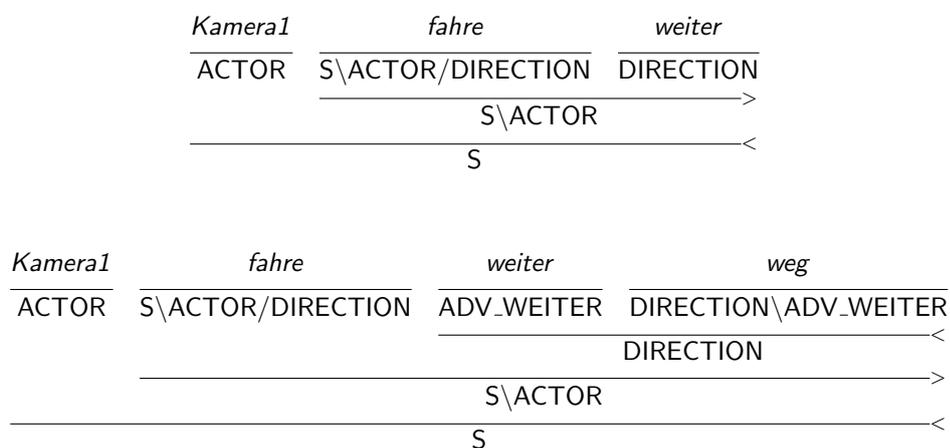


Abbildung 5.13: Ableitung eines Komparativs unter Verwendung von *weiter*.

Weiter, in der Verwendung als Adjektiv, benötigt eine Richtung, die es als Füllwort ergänzt und in eine Komparationsform wandelt. In *Kamera1 fahre weiter weg* wandelt *weiter* die Richtung *weg* in eine gesteigerte Form. Abbildung 5.13 (unten) entwickelt die Ableitung des Beispielsatzes. Die Kategorie für die Richtung *weg* wird um *DIRECTION\ADV_WEITER* ergänzt, um zu verdeutlichen, dass *weg* genau dann einer *DIRECTION* zugeordnet werden kann, wenn ihm ein *ADV_WEITER* vorsteht. Abbildung 5.14 zeigt die logische Form dieser Anweisung. Die Richtungsbeschreibung *DIRECTION* wird hierbei durch eine Eigenschaft *prop* ergänzt, die das Wort *weiter* aufnimmt.

```

@w1:action(joystick ^
  <VERB>fahre ^
  <ACTOR>(w0:actor ^ Kamera1 ^
    <num>sg-X ^
    <gender>genderFem-X) ^
  <DIRECTION>(x1:direction ^
    <dir>weg ^
    <prop>(w2 ^ weiter)))

```

Abbildung 5.14: Logische Form von *Kamera1 fahre weiter weg*.

Näher erzeugt, vergleichbar mit *weiter*, ebenfalls eine Komparation. Sie kann aber nur im Zusammenhang mit *ran* verwendet werden. Grammatikalisch wird es wie das Adjektiv *weiter*

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

behandelt. Beide Adjektive *weiter* und *näher* ergänzen Sätze und drücken dabei aus, dass eine bestimmte Bewegung in der zuvor definierten Richtung fortgesetzt werden soll. So wird die Kategorie des Wortes *weg* durch die logische Form $\models @_{direction}(dir_D \wedge prop_W)$ und die Kategorie von *weiter* durch $\models @(X)$ ergänzt.

5.2.4 Richtung durch ortsbeschreibende Adverben

Durch die Präposition *nach*, gefolgt von einem ortsbeschreibenden Adverb, kann ebenfalls eine Richtung ausgedrückt werden. Als Beispiel dafür dient der in Abbildung 5.15 dargestellte Satz *Kamera1 schwenke nach rechts*. Für die Ableitung sind zwei Kategorien POSITION und DIRECTION/POSITION nötig, um in richtiger Kombination wieder eine Kategorie DIRECTION zu erzeugen.

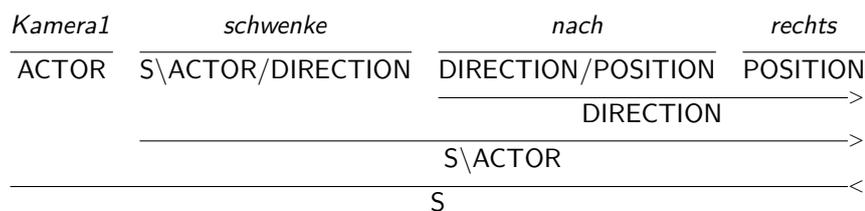


Abbildung 5.15: Ableitung eines Joystick-Kommandos mit Hilfe der CCG.

Die ortsbeschreibenden Adverben *rechts* und *links* sowie *oben* und *unten* werden der Kategorie POSITION zugeordnet. Die Kategorie für die Präposition *nach* legt fest, dass sie zusammen mit einem nachfolgenden ortsbeschreibenden Adverb eine Richtung DIRECTION beschreibt. Ein zulässiger Satz kann genau dann abgeleitet werden, wenn vor dem Verb ein Adressat ACTOR und hinter dem Verb eine Richtung DIRECTION angegeben ist. In Abbildung 5.15 wird das anhand des Verbs *schwenke* demonstriert. Abbildung 5.16 zeigt die von der OpenCCG generierte logische Form des Joystick-Kommandos aus Abbildung 5.15.

Ein Joystick-Kommando enthält drei Konstituenten: VERB, ACTOR und DIRECTION. VERB enthält nur das verwendete Verb *schwenke*, ACTOR die angesprochene Kamera und DIRECTION die angeforderte Richtung. Um diese Form zu generieren, wird die Kategorie des Verbs *schwenken* um die logische Ableitung $\models @_{action}(joystick \wedge VERB_Z \wedge ACTOR_X \wedge DIRECTION_Y)$ erweitert. Diese beschreibt eine Joystick-Aktion, die durch die Nominale VERB, ACTOR und DIRECTION ergänzt wird. Die logische Form für die Kategorie von *Kamera1* wurde oben bereits vorgestellt.

```

@w1:action(joystick ^
  <VERB>schwenke ^
  <ACTOR>(w0:actor ^ Kamera1 ^
    <num>sg-X ^
    <gender>genderFem-X) ^
  <DIRECTION>(x1:direction ^
    <prep>nach ^
    <position>(w3 ^ rechts)))

```

Abbildung 5.16: Logische Form eines Joystick-Kommandos.

Die Kategorie POSITION wird in Ihrer logischen Form durch $\models @ (P)$ repräsentiert. Die Präposition *nach* erzeugt eine Richtungskategorie DIRECTION. Die logische Form für diese Kategorie wird deshalb durch $\models @_{direction}(prep_N \wedge position_P)$ beschrieben.

Berücksichtigt man an dieser Stelle die adjektivische Form von *weiter*, muss die logische Form als $\models @_{direction}(prep_N \wedge cont_W \wedge position_P)$ definiert werden. Aus *fahre nach rechts* wird so die umgangssprachliche Form *fahre weiter nach rechts*.

5.2.5 Unbestimmte Quantität

Eine unbestimmte Quantitätsangabe ergänzt die vorhandene Richtungsangabe und bestimmt die Intensität der Bewegung. Wird ein reduzierendes Quantitätswort ausgesprochen, erwartet der Nutzer entsprechend weniger Bewegung als wenn keine Quantitätsangabe erfolgt. Für ein vermehrendes Wort gilt invers Äquivalentes. Typische Quantitätswörter sind: *etwas*, *ein wenig*, *deutlich* etc.

Quantitätsbegriffe können sowohl eigenständig als auch an die jeweilige Richtung gebunden in der Grammatik definiert werden. In der logischen Form müssen sie dann, für den Fall der Eigenständigkeit, explizit aufgeführt werden. Sind sie jedoch der jeweiligen Richtung untergeordnet, werden sie auch in der logischen Form dieser Richtung ergänzend angegeben. Von der Aussage her unterscheiden sich die beiden Verwendungen durch den jeweiligen Bezug. Entweder quantifizieren sie das Verb oder die Richtung. Im ersten Fall bezieht sich die Quantität auf *fahren*. Die Kernaussage liegt hierbei auf *fahre ein bisschen*. Im zweiten Fall bezieht sich die Quantität auf die Richtung. Die Aussage ergibt sich zu *ein bisschen nach rechts*. Um klar den Bezug der angegebenen Quantität erkennen zu können, werden die Quantitätsbegriffe hier der jeweiligen Richtung untergeordnet. Somit enthält die logische Form einer Richtungsphrase – sofern formuliert – auch eine Quantitätsangabe.

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

Zur Abbildung dieser Begriffe werden unterschiedliche Kategorien angelegt. Für atomare (einwortige) Quantitätsbegriffe, wie *etwas*, *deutlich* oder *weit*, wird eine eigene Kategorie DIRECTION/DIRECTION angelegt. Abbildung 5.17 zeigt oben die Ableitung des einwortigen Quantitätsbegriffs *etwas* im Rahmen der Phrase *etwas nach rechts*.

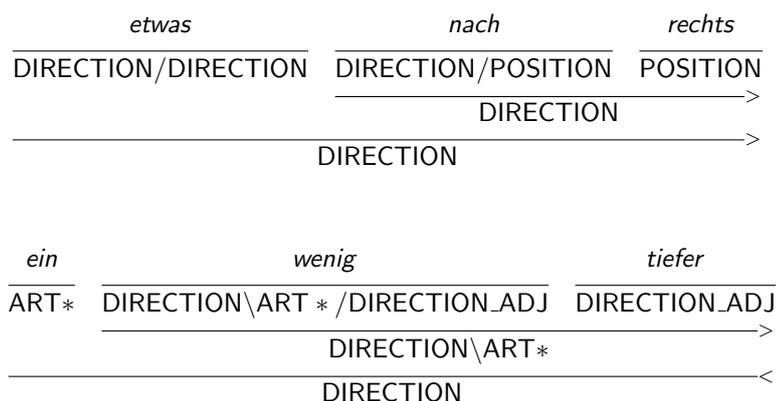


Abbildung 5.17: Zusammengesetzte Quantitätsangabe. Aus Platzgründen wurde ART* eingeführt. Dieser steht für die Kategorie $ART_{undef,caseNom,genderNeut,sg}$, die die *features* *def*, *case*, *gender* und *num* berücksichtigt.

Zusammengesetzte Ausdrücke wie *ein wenig* oder *ein bisschen* müssen durch Hilfskategorien abgebildet werden. Diese Hilfskategorien erlauben es, die zusammengesetzten Ausdrücke als Quantitätsangaben zu erkennen. Die Weiterverarbeitung geschieht wie bei den direkten Quantitätsbegriffen. Den oben genannten Ausdrücken geht jeweils ein singulärer unbestimmter Nominativ-Artikel voraus. Abbildung 5.17 zeigt ein Beispiel (unten) für die Ableitung zusammengesetzter Ausdrücke.

Um einwortige Quantitätsbegriffe in die logische Form zu überführen, wird die Kategorie DIRECTION/DIRECTION um den Ausdruck $\models @(dir_D \wedge quant_Q)$ ergänzt. Dabei wird eine Richtung durch ein Richtungswort und einer Quantität beschrieben. Für zusammengesetzte Begriffe wird obige Form durch die Angabe des Artikels erweitert und ergibt so $\models @(direction_D \wedge quant_Q \wedge art_A)$.

Mit diesen Kategorien können jetzt Sätze, wie *Kamera1 schwenke ein wenig höher* oder *Kamera1 fahre etwas zurück* abgeleitet und logisch interpretiert werden.

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

QUANTITY\NUMBER folgen. Der Repräsentant für das Zahlwort ist im vorgestellten Beispiel 3. Der Begriff *Einheiten* dient der Identifikation der bestimmten Quantität und wird deshalb der Kategorie QUANTITY\NUMBER zugeordnet. Abbildung 5.21 zeigt den vollständigen logischen Ausdruck für den Beispielsatz aus Abbildung 5.20.

```
@w1:action(joystick ^
  <VERB>fahre ^
  <ACTOR>(w0:actor ^ Kamera1 ^
    <num>sg-X ^
    <gender>genderFem-X) ^
  <DIRECTION>(x1:direction ^
    <prep>nach ^
    <dir>(w5 ^ rechts)) ^
  <QUANTITY>(w3:quantity ^ Einheiten ^
    <number>(w2:number ^ 3)))
```

Abbildung 5.21: Logische Form eines Joystick-Kommandos in Verbindung mit bestimmter Quantität.

Für die logische Betrachtung muss die Kategorie des Verbs *fahre* um den Ausdruck $\models @_{action}(joystick \wedge VERB_Z \wedge ACTOR_X \wedge DIRECTION_D \wedge QUANTITY_Q)$ ergänzt werden. Sie beschreibt eine Joystick-Aktion, die durch die Nominalen *VERB*, *ACTOR*, *DIRECTION* und *QUANTITY* beschrieben wird. Die Kategorie QUANTITY\NUMBER wird um die Vorschrift $\models @_{quantity}(Q \wedge number_Y)$ und die Kategorie NUMBER wird um $\models @_{number}(X)$ erweitert.

5.3 Objektbezogene Kommandos

Die *Einstellungsfindung* ist ein Prozess zwischen Regie und Kamera, mit dem Ziel bestimmte Objekte in möglichst ästhetischer Art und Weise auf dem Bildschirm zu platzieren. Dabei wird die Darstellung einzelner Objekte im Bild angepasst. Im Folgenden werden Kommandos für diese Bildmodifikationen vorgestellt.

5.3.1 Bildausschnittsbestimmende Kommandos

Die grundsätzliche Orientierung der Kamera und der dazu passende Öffnungswinkel der Linse kann mit Hilfe bildausschnittsbestimmender Kommandos festgelegt werden. Die Kamera und der Öffnungswinkel werden so eingestellt, dass sie die entscheidenden bzw. gewünschten Objekte zeigen. Ausgehend von dieser Grundposition kann dann die Einstellung verfeinert und dem

individuellen Empfinden nach angepasst werden. Abbildung 5.22 repräsentiert beispielhaft zwei Kommandos der hier beschriebenen Kommando-Gruppe.

```
Kamera1 zeige den Moderator
Kamera1 zeige mir den Moderator und das Objekt
```

Abbildung 5.22: Beispiele für ausschnittsbestimmende Kommandos.

Die Kommandos sind aus dem im Prototypen HULICO (s. Kapitel 7.1) verwendeten Test-Szenario entnommen. In diesem Szenario gibt es einen *Moderator* sowie ein weiteres Kulissenobjekt *Objekt*. Das Beispiel in Abbildung 5.22 oben gibt der Präsentation des Moderators den Fokus im Bild. Im zweiten Beispiel (unten) soll neben dem Moderator auch das Objekt im Bild zu sehen sein. Abbildung 5.23 zeigt die Struktur dieser Kommandos.

```
[SHOW-CMD] := [ACTOR] [zeigen] [ENUMERATION]
[SHOW-CMD] := [ACTOR] [zeigen] [REF] [ENUMERATION]
```

Abbildung 5.23: Struktur der bildausschnittsbezogenen Kommandos.

Vor dem Verb *zeige* steht der **ACTOR**. Dem Verb folgt entweder ein Reflexivpronomen **REF** und eine **ENUMERATION** oder eine **ENUMERATION** ohne Zusatz. **REF** gibt an, wem die nachfolgende Enumeration gezeigt werden soll. Das ist im Studio meist der Regisseur. Trotzdem ist diese Formulierung üblich. Der Begriff **ENUMERATION** steht für ein oder mehrere aufgezählte Objekte.

Die Aufzählung **ENUMERATION** ist durch eine Aneinanderreihung von Nominalphrasen charakterisiert. Die Konjunktion *und* verbindet diese Reihung. Abbildung 5.24 zeigt die Ableitung des vorgenannten, unteren Beispielsatzes *Kamera1 zeige mir den Moderator und das Objekt*.

Die Kategorie des Verbs *zeige* kann genau dann abgeleitet werden, wenn vor ihr eine Kategorie vom Typ **ACTOR** und nach ihr eine Kategorie des Typs **NP** sowie eine Enumeration **ENUM** folgen. **NP** steht für eine Nominalphrase und muss im Dativ stehen. Nominalphrasen bestehen im Normalfall aus einem Nomen **N** und einem Artikel **NP/N**. In dem hier präsentierten Beispielsatz könnte also anstatt *mir* auch *dem Chef-Regisseur* oder jede andere Person stehen. Durch die Angabe von **NP** wird der Normalfall dargestellt. *mir* bildet hier eine Ausnahme und wird direkt der Kategorie **NP** zugeordnet. So ist *mir* ein Pronomen, das für eine Nominalphrase steht. **ENUM** kann sowohl durch eine Nominalphrase **NP** als auch durch eine Aneinanderreihung von Nominalphrasen, getrennt durch die Konjunktion *und*, generiert werden. Die Kategorie von *und* muss dafür mit **ENUM\NP/NP** und **ENUM\ENUM/NP** ergänzt werden. Der erste Fall generiert die Kategorie

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

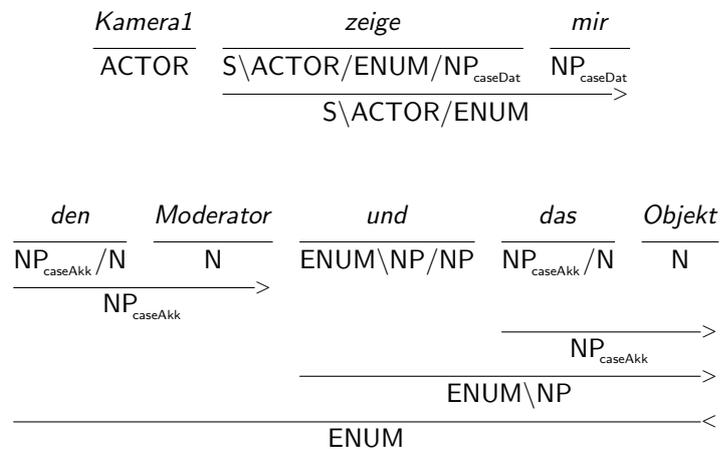


Abbildung 5.24: Ableitung eines bildausschnittsbestimmenden Kommandos.

ENUM für die ersten zwei durch *und* getrennten Nominalphrasen, der zweite Fall ergänzt diese Aufzählung durch weiter folgende mit *und* getrennte Nominalphrasen. Abbildung 5.25 zeigt den logischen Ausdruck des oben eingeführten Beispielsatzes.

```

@w1:action(show ^
  <VERB>zeige ^
  <ACTOR>(w0:actor ^ Kamera1 ^
    <num>sg-X ^
    <gender>genderFem-X) ^
  <ENUMERATION>(x1:enumeration ^
    <item>(w4:obj ^ Moderator ^
      <num>sg-X ^
      <gender>genderMasc-X ^
      <art>den) ^
    <item>(w7:obj ^ Objekt ^
      <num>sg-X ^
      <gender>genderNeut-X ^
      <art>das)) ^
  <REFP>(w2 ^ mir ^
    <case>caseDat-X ^
    <num>sg-X ^
    <pers>1st-X))
  
```

Abbildung 5.25: Logische Form eines bildausschnittsbestimmenden Kommandos.

Der Kategorie des Verbs *zeige* wird die logische Ergänzung $\models @_{action}(show \wedge VERB_X \wedge$

$ACTOR_Y \wedge ENUMERATION_Z \wedge REFP_W$) zugeordnet. Dies bezeichnet eine Zeige-Aktion, die durch die Nominale *VERB*, *ACTOR*, *ENUMERATION* und *REFP* beschrieben wird. *VERB* beschreibt das verwendete Verb, *ACTOR* die angesprochene Kamera, *ENUMERATION* die sichtbaren Objekte im Bild in Form einer Aufzählung und *REFP* beschreibt, wem die Objekte gezeigt werden sollen. Bei der Verwendung ohne Reflexivpronomen wird das Nominal $REFP_W$ weggelassen. Für die logische Ableitung der Kategorie $ENUM \setminus NP / NP$ werden diese durch $\models @_{enumeration}(item_X \wedge item_Y)$ und der Ausdruck $ENUM \setminus ENUM / NP$ durch die logische Ableitung $\models @_{enumeration}(E \wedge item_Z)$ erweitert. Für die logische Behandlung einer Nominalphrase wird die Kategorie des Artikels NP / N durch den Ausdruck $\models @(art X)$ ergänzt. Der Artikel wird so durch ein Nominal *art* beschrieben und kann in eine bestehende Struktur übernommen werden. Die Kategorie des Nomens *N* wird durch $\models @_{obj}(X)$ ergänzt. Die Kategorie von *mir* wird durch $\models @(X)$ ergänzt.

5.3.2 Kommandos zur Positionierung von Objekten im Bild

Änderungen im Bild lassen sich durch Verschiebungen der im Bild sichtbaren Objekte beschreiben. Das erleichtert die Formulierung von Änderungen im Bild und abstrahiert die einzelnen Achsen des Kamerasystems. So kann ohne Rücksicht auf die zu verwendeten Achsen eine Änderung im Bild angewiesen werden. In Abbildung 5.26 werden einige Beispiele aufgezählt, die eine Änderung des Bildes auslösen.

```
Kamera1 setze den Moderator mittig
Kamera1 setze das Objekt auf die linke Seite
Kamera1 setze den Moderator auf 25%
Kamera1 setze den Moderator weiter nach links
```

Abbildung 5.26: Beispiele zum Verschieben von Objekten im Bild anhand von absoluter (Zeilen 1-3) und relativer (Zeile 4) Positionierung.

Typischerweise werden Objekte links, zentral oder auf der rechten Seite positioniert. Ein guter Anhaltspunkt für diese Anweisungen sind die Grenzen der sogenannten Drittel-Regel. Neben dieser absoluten Positionierung, kann ein Objekt auch relativ positioniert werden. Dabei wird das Objekt in die angesprochene Richtung verschoben. Die ersten drei genannten Beispiele in Abbildung 5.26 sind mit absoluter Positionierung gewählt. Das vierte Beispiel beschreibt eine relative Positionierung. Das Positionieren von Objekten wird mit Hilfe des Verbs *setzen* beschrieben. Abbildung 5.27 zeigt die Struktur der möglichen Anweisungen.

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

[COMPOSE-CMD] := [ACTOR] [setzen] [OBJ_AKK] [POSITION]
 [COMPOSE-CMD] := [ACTOR] [setzen] [OBJ_AKK] [DIRECTION]

Abbildung 5.27: Struktur des Compose-Kommandos mit absoluter (POSITION) und relativer (DIRECTION) Positionierung.

Strukturell folgt dem Verb ein Akkusativ-Objekt OBJ_AKK, dem die jeweilige Seite POSITION bzw. Richtung DIRECTION folgt. Folgt dem Objekt eine Richtung, so wird eine Verschiebung des genannten Objekts im Bild gewünscht. Dabei steht POSITION für eine absolute und DIRECTION für eine relative Positionierung. Abbildung 5.28 zeigt die Ableitung des Beispielsatzes *Kamera1 setze den Moderator mittig*.

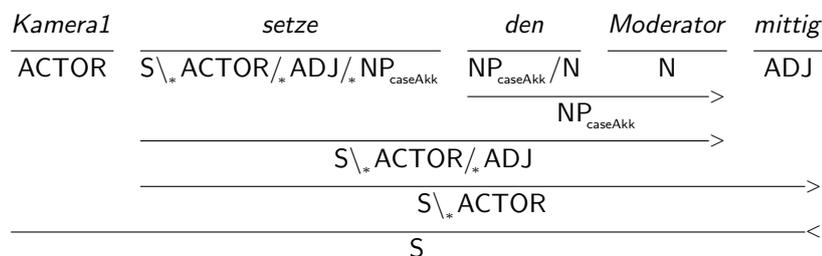


Abbildung 5.28: Ableitung eines Compose-Kommandos.

Für die Kategorie des Verbs *setze* wird hierbei festgelegt, dass diesem Wort ein Akteur ACTOR vorausgeht und sowohl eine Nominalphrase NP wie auch ein Adjektiv ADJ folgen. Die Nominalphrase muss hierbei im Akkusativ stehen. Für den gegebenen Satz wird die Nominalphrase durch *den Moderator* gebildet. Das Adjektiv ist hier *mittig*.

Für die logische Form wird die Kategorie von ACTOR, wie in den vorstehenden Kapiteln bereits vorgestellt, übernommen. Gleiches gilt für die Kategorie NP. Die Kategorie des Verbs wird durch $\models @_{action}(composeObjectAdj \wedge VERB_V \wedge ACTOR_X \wedge PATIENT_Y \wedge POSITION_P)$ ergänzt. Diese Form beschreibt eine Compose-Aktion, die durch die Nominalphrase *VERB*, *ACTOR*, *PATIENT* und *POSITION* beschrieben wird. Abbildung 5.29 zeigt die generierte logische Form für den obigen Beispielsatz.

Das Adjektiv kann direkt in die logische Form übernommen werden und wird deshalb durch $\models @_{prop}(X)$ ergänzt. Das Nominal PATIENT ergibt sich aus der Satzstruktur und übernimmt die logische Formulierung einer Nominalphrase (hier: *den Moderator*). Anweisungen dieser Art verändern eine Einstellung, ohne dabei typische Fahrbefehle zu geben. Es obliegt der Kamera, wie die allgemeine Änderungsanweisung umgesetzt wird.

```

@w1:action(composeObjectAdj ^
  <VERB>setze ^
  <ACTOR>(w0:actor ^ Kamera1 ^
    <num>sg-X ^
    <gender>genderFem-X) ^
  <PATIENT>(w3:obj ^ Moderator ^
    <num>sg-X ^
    <gender>genderMasc-X ^
    <art>den) ^
  <POSITION>(w4:prop ^ mittig))

```

Abbildung 5.29: Logische Form eines Compose-Kommandos.

5.3.3 Kommandos zur Raumänderung von Objekten

Die Verwendung der raumändernden Kommandos gibt der Regie die Möglichkeit, den Abstand zwischen Moderator und Bildrand bzw. allgemein zwischen Objekten und Bildrand zu beeinflussen. Die Einleitung dieses Kommandos geschieht über die Verben *geben* und *nehmen*. Abbildung 5.30 zeigt einige Beispiele solcher Kommandos.

```

Kamera1 gib dem Moderator mehr Luft
Kamera1 gib dem Moderator oben mehr Luft
Kamera1 nimm dem Moderator oben etwas Luft

```

Abbildung 5.30: Beispiele zum Ändern von Objekt-Eigenschaften.

Mit *geben* kann sowohl eine Vergrößerung als auch eine Verkleinerung des Abstandes zum Rand gefordert werden. Üblicherweise vergrößert das Wort *geben* den Rand. Es kann allerdings auch, in Verbindung mit dem Quantitätswort *weniger*, eine reduzierende Wirkung haben. Das Verb *nehmen* verringert immer den Abstand. Wird keine Position genannt, gilt das Kommando allgemein für das referenzierte Objekt. Durch direkte Nennung der Position kann die Seite, an der der Raum zwischen Objekt und Rand geändert werden soll, benannt werden. Abbildung 5.31 zeigt die Struktur dieses Kommandos.

```
[MODIFY_OBJ-CMD] := [ACTOR] [geben|nehmen] [OBJ_DAT] [OBJ_AKK]
```

Abbildung 5.31: Struktur der Kommandos zum Ändern von Objekt-Eigenschaften.

Den Verben *geben* bzw. *nehmen* geht der ACTOR voraus und ihnen folgt ein Objekt im Dativ OBJ_DAT und ein Objekt im Akkusativ OBJ_AKK. Wird eine bestimmte Position benannt, so

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

beschreibt diese Position die gewünschte Anwendungsseite genauer. Die Positionsbestimmung wird deshalb dem Akkusativ-Objekt zugeordnet. Die Ableitung des Kommandos zeigt Abbildung 5.32 anhand des Beispielsatzes *Kamera1 gib dem Moderator oben mehr Luft*.

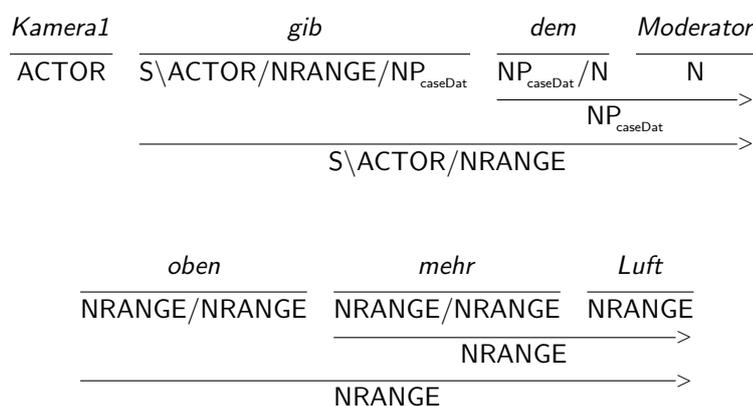


Abbildung 5.32: Ableitung eines modifyObject-Kommandos.

Im oberen Abschnitt wird die Phrase *Kamera1 gib dem Moderator* abgeleitet. Im unteren Teil folgt die Phrase *oben mehr Luft*. Der obere Teil endet deshalb mit dem Konstrukt $\text{S}\backslash\text{ACTOR}/\text{NRANGE}$. Der untere Teil wird zu NRANGE abgeleitet. Auf die vollständige Ableitung wird in der Abbildung verzichtet. *Kamera1* wird wie in jedem Kommando der Kategorie ACTOR zugeordnet. Die Phrase *dem Moderator* wird der Dativ-Nominalphrase NP zugewiesen. Für die Phrase *oben mehr Luft* werden die Adverbien *oben* und *mehr* der Kategorie von *Luft* NRANGE untergeordnet. Sie beschreiben die Luft näher und generieren deshalb aus einer nachfolgenden Kategorie NRANGE wieder eine Kategorie NRANGE . In CCG-Notation wird das durch $\text{NRANGE}/\text{NRANGE}$ ausgedrückt. Das Verb *geben* wird einer Kategorie zugeordnet, die vorausgehend einen ACTOR und nachfolgend eine Nominalphrase und einen NRANGE -Ausdruck benötigt. Abbildung 5.33 zeigt die logische Form des obigen Beispielsatzes.

Für die logische Form können die vorhergehenden Ergänzungen für den Akteur ACTOR und die Nominalphrase NP übernommen werden. Die Kategorie von *Luft* NRANGE wird durch $\models @_{\text{range}}(X)$ und *geben* durch den Ausdruck $\models @_{\text{action}}(\text{modifyObject} \wedge \text{VERB}_V \wedge \text{ACTOR}_X \wedge \text{PATIENT}_Y \wedge \text{PROP}_P)$ ergänzt. Es handelt sich also um eine *modifyObject*-Aktion, die durch die Nominale VERB , ACTOR , PATIENT und PROP ergänzt wird. Das Adverb *oben* wird durch $\models @(\text{pprop}_x)$ erweitert. pprop bezeichnet dabei eine seitenbeschreibende Eigenschaft (Position-Property). *mehr* wird durch $\models @(\text{qprop}_x)$ ergänzt. Letzteres bezeichnet eine Quantitätseigenschaft.

```

@w1:action(modifyObject ^
  <VERB>gib ^
  <ACTOR>(w0:actor ^ Kamera1 ^
    <num>sg-X ^
    <gender>genderFem-X) ^
  <PATIENT>(x1:object ^
    <art>dem ^
    <object>(w3:obj ^ Moderator ^
      <num>sg-X ^
      <gender>genderMasc-X)) ^
  <PROP>(w6:range ^ Luft ^
    <num>sg-X ^
    <gender>genderFem-X ^
    <pprop>oben ^
    <qprop>mehr))

```

Abbildung 5.33: Logische Form eines modifyObject-Kommandos.

5.4 Konditional-Kommandos

Die Feinjustierung von Einstellungen erfordert oft kleinste Änderungen am Kamerabild, die allein mit Joystick-Anweisungen nur sehr schwer erreicht werden. Nur ein explizites Benennen der Verfehlung stellt sicher, dass ausschließlich das störende Element aus dem Bild entfernt wird. Die Regie nennt eine Bewegung und gibt gleichzeitig an, bis zu welchem Zustand diese Bewegung ausgeführt werden soll. Diese Kommandos werden im Weiteren als Konditionalkommandos bezeichnet.

Ein kleines Beispiel soll die Verwendung dieses Kommandos verdeutlichen: Eine Einstellung sei weitestgehend fertig justiert. Im Bild ist dabei allerdings ein Dekorations-Element angeschnitten zu sehen. Mit den bisher vorgestellten Anweisungen kann das System dieses Problem nicht zufriedenstellend lösen. Mit einem Konditional-Kommando kann die Intention der Bewegung dem Steuer-System mitgeteilt werden, so dass sichergestellt werden kann, dass die Bewegung mit dem Ausschluss des störenden Elements beendet wird. Abbildung 5.34 zeigt mögliche Beispiele für dieses Kommando.

```

Kamera1 fahre nach links bis das Objekt nicht mehr zu sehen ist.
Kamera1 schwenke nach rechts bis das Objekt zu sehen ist.

```

Abbildung 5.34: Beispiele für Konditional-Kommandos.

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

Eine typische Anwendung dieser Kommandos ist das Erreichen der Sichtbarkeit bzw. der Nicht-Sichtbarkeit eines bestimmten Objekts im Bild. Dabei soll im oberen Beispiel ein Objekt aus dem Bild ausgeschlossen, im unteren explizit eingeschlossen werden. Die Struktur dieses Kommandos teilt sich auf zwei Teilsätze auf, die durch Konjunktionen wie *bis* oder *sodass* verbunden werden. Abbildung 5.35 zeigt die Struktur des Kommandos.

[CONDITIONAL-CMD] := [ACTOR] [ACTION] [bis|sodass] [STATE]

Abbildung 5.35: Struktur der Konditional-Kommandos.

Vor dem Bindewort wird die jeweilig durchzuführende Grundaktion beschrieben. Diese Aktion ist eine bestimmte Joystick-Anweisung. Nach dem Bindewort wird der gewünschte Zielzustand beschrieben. In Abbildung 5.35 ist die Aktion durch **ACTION** und der Zielzustand durch **STATE** beschrieben. Am Satzanfang steht, wie bereits in Abbildung 5.1 dargestellt, der Adressat des Kommandos **ACTOR**.

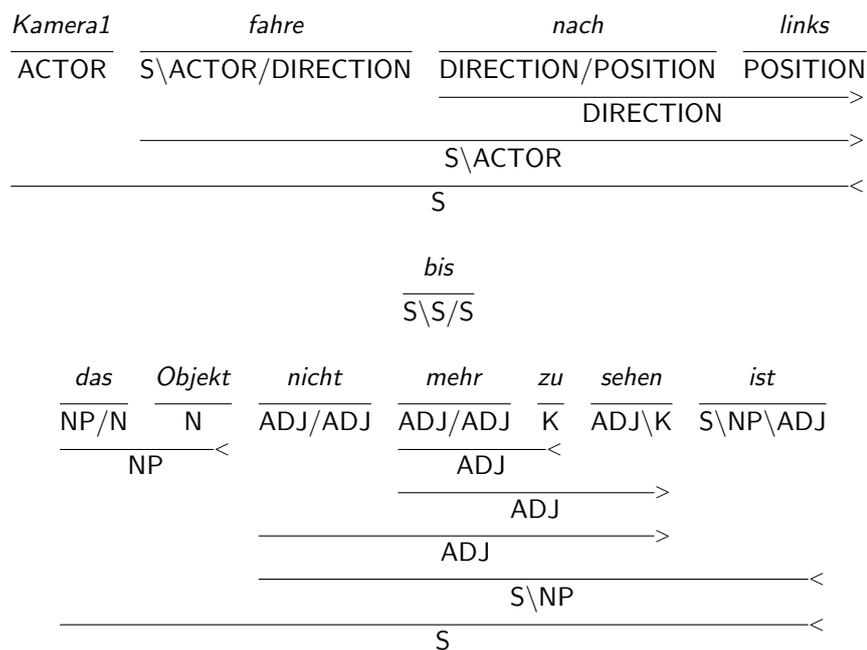


Abbildung 5.36: Ableitung eines Konditional-Kommandos.

Abbildung 5.36 entwickelt die Ableitung des Beispielsatzes *Kamera1 fahre nach links bis das Objekt nicht mehr zu sehen ist*. Der erste Teil des Satzes ergibt sich wie in Kapitel 5.2 bereits gezeigt. Das Bindewort *bis* wird einer Kategorie zugeordnet, die aus einem vorhergehenden

und einem nachfolgenden Satz einen weiteren Satz erzeugt $S \setminus S / S$. Der zweite Satzteil basiert auf dem Verb *sein* in der dritten Person Singular (*ist*). Dieser legt fest, dass vorausgehend eine Nominalphrase und ein Adjektiv erforderlich sind. Durch dieses Konstrukt wird einem Objekt, das durch eine Nominalphrase beschrieben ist, eine bestimmte Eigenschaft zugeordnet. Im vorliegenden Beispielsatz wird diese Eigenschaft durch das Adjektiv *sichtbar* in der Infinitiv-Form *zu sehen* ersetzt und zusätzlich negiert (*nicht mehr*). Die Phrase *zu sehen* wirkt wie ein Adjektiv. Deshalb wird *sehen* einer Kategorie zugeordnet, die zusammen mit der Konjunktion *zu* eine Kategorie ADJ erzeugt. Sowohl die Negation *nicht* als auch das Adverb *mehr* werden Teil dieser Kategorie. Die jeweiligen Kategorien ergeben sich deshalb zu ADJ/ADJ.

Der in Abbildung 5.36 vorgestellte Beispielsatz könnte deutlich einfacher durch *... das Objekt unsichtbar ist* formuliert werden. Das Adjektiv *unsichtbar* wird in diesem Satz direkt der Kategorie ADJ zugeordnet und somit die Satzstruktur deutlich verkürzt. Das gewählte Beispiel wurde jedoch so in der Regie verwendet und deshalb hier vorgestellt. Abbildung 5.37 zeigt die logische Form des Beispielsatzes.

```
@w4:condition(condition ^
  <ACTION>(w1:action ^ joystick ^
    <VERB>fahre ^
    <ACTOR>(w0:actor ^ Kamera1 ^
      <num>sg-X ^
      <gender>genderFem-X) ^
    <DIRECTION>(x1:direction ^
      <prep>nach ^
      <position>(w3 ^ links))) ^
  <CONDITION>(w11:state ^ ist ^
    <PATIENT>(w6:obj ^ Objekt ^
      <num>sg-X ^
      <gender>genderNeut-X ^
      <art>das) ^
    <PROP>(w10 ^ sehen ^
      <case>caseInf-A ^
      <cont>mehr ^
      <neg>nicht)))
```

Abbildung 5.37: Logische Form eines Konditional-Kommandos.

Die Kategorie der Konjunktion *bis* wird durch den Ausdruck $\models @_{condition}(condition \wedge ACTION_A \wedge CONDITION_C)$ erweitert. So wird dieser Satz als *condition* mit den Nomi-

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

nenen ACTION und CONDITION markiert. Für die logische Form des Teilsatzes ACTION wurden die Grundelemente bereits in den vorhergehenden Kapiteln definiert. Der Teilsatz CONDITION wird wie folgt verarbeitet: Für die Kategorie des Verbs *ist* wird der Zustandssatz durch $\models @_{state}(state \wedge PATIENT_P \wedge PROP_T)$ beschrieben. Diese Form beschreibt einen Zustand *state* mit den Nominalen *VERB* und *PATIENT*. Das Adverb *mehr* wird durch $\models @(cont_Y)$, das Adverb *nicht* durch $\models @(neg_X)$ erfasst. Die Kategorie des Verbs *sehen* wird durch $\models @(X)$ ergänzt.

5.5 Speichern-Kommando

Das Speichern-Kommando bietet die Möglichkeit, die aktuelle Situation zu speichern, um beispielsweise am nächsten Tag oder nach einer Pause von der aktuell gegebenen Position fortzufahren. Außerdem können so potentiell in Frage kommende Einstellungen für eine spätere Diskussion festgehalten werden. Im Kontext des Systems kommt diesem Kommando eine Sonderrolle zu. Es hat keinen Bezug zur Bewegung. Dennoch haben die Tests mit dem hier beschriebenen System gezeigt, dass dieses administrative Kommando seine Berechtigung hat. Es erleichtert die Arbeit mit dem System und ermöglicht es Fortschritte exakt zu protokollieren. Im Hinblick auf die Verwendbarkeit im Rahmen der *Sendekontrolle*, erlaubt es dieses Kommando Eigenschaften des Bildes zu speichern, die bei einer Wiederanfahrt mit berücksichtigt werden können. Die Beispiel-Kommandos in Abbildung 5.38 zeigen, wie das Speichern verbal eingeleitet wird.

```
Kamera1 speicher die Einstellung  
Kamera1 speicher die Einstellung als Shot1  
Kamera1 speicher die aktuelle Einstellung als Shot2
```

Abbildung 5.38: Beispiele für das Speichern der aktuellen Position.

Das Speichern-Kommando kann sowohl mit einem als auch mit zwei Objekten gebildet werden kann. Im ersten Fall wird kein, im zweiten Fall wird das Ziel der Aktion explizit bestimmt. Abbildung 5.39 zeigt die Struktur oben ohne und unten mit expliziter Angabe des Speicherziels.

```
[SAVE-CMD] := [ACTOR] [speichern] [WHAT]  
[SAVE-CMD] := [ACTOR] [speichern] [WHAT] [TARGET]
```

Abbildung 5.39: Struktur des Speichern-Kommandos.

In der indikativen Verwendung geht dem Verb *speichern* die Bezeichnung des Akteurs **ACTOR** voraus. Der Akteur ist der Empfänger des Kommandos. Im vorliegenden Fall ist das immer die angesprochene Kamera. Das Bezugsobjekt **WHAT** folgt dem Verb. Es gibt an, was gespeichert werden soll. In allgemeiner Verwendung können Entitäten jeglicher Form gespeichert werden. Es kommen z.B. Farben, Formen oder auch bestimmte Bild-Kompositionen in Frage. Im Kontext eines Kamerasystems ist das Bezugsobjekt immer die aktuelle Kameraposition und damit die jeweilige Einstellung der Kamera. *Speichern* wird deshalb in Verbindung mit verbaler Kamerakontrolle immer in der Kombination mit dem Bezugsobjekt *die Einstellung* Verwendung finden. In der logischen Form könnte es auch weggelassen werden. Der Vollständigkeit halber wird dennoch der Bezug aufgeführt, um einerseits grammatikalisch korrekte Sätze zu fordern und andererseits die allgemeine Verwendbarkeit der Kommandostruktur zu erhalten. Wird das Kommando mit Angabe eines Speicherziels **TARGET** verwendet, so wird dieses dem Bezugsobjekt **WHAT** angehängt. Das Speicherziel wird mit der Konjunktion *als* eingeleitet. Der Konjunktion folgt ein Name, der das tatsächliche Ziel der Speicheraktion angibt. In CCG bekommt *speichern* zwei Kategorien zugeteilt. In Abbildung 5.40 sind diese dargestellt.

$$\begin{aligned} \textit{speichern} &:= S \setminus \text{ACTOR} / \text{NP} [\textit{caseAkk}] \\ &:= S \setminus \text{ACTOR} / \text{KP} / \text{NP} [\textit{caseAkk}] \end{aligned}$$

Abbildung 5.40: Kategorien für das Verb *speichern*.

Die erste Alternative beschreibt das Speichern ohne, die zweite das Speichern mit Speicherziel. Die Nominalphrase NP steht für das Bezugsobjekt (**WHAT**, s. Abbildung 5.39) und wird durch einen Artikel, gefolgt von einem Nomen, erzeugt. In diesem Fall muss die Nominalphrase im Akkusativ ausgedrückt sein. Um Sätze mit Angabe des Speicherziels ableiten zu können, wird eine zusätzliche komplexe Kategorie eingeführt (s. Abbildung 5.40 unten). Sie fordert zur bereits bestehenden Kategorie nachfolgend ein Konstrukt aus Konjunktion und Nomen. Dieses Konstrukt wird in der Abbildung 5.40 als *Konjunkionalphrase* KP bezeichnet. Die CCG ist rechts-bindend, d.h. die Ableitung einer komplexen Kategorie erfolgt von rechts nach links. Für das Verb *speichern* bedeutet das, dass die in der komplexen Kategorie ganz rechts notierte NP zuerst durch die dem Verb folgende Nominalphrase kompensiert wird. Diesem Konstrukt muss eine KP folgen. KP wird deshalb vor der Nominalphrase eingefügt. Abbildung 5.41 zeigt die Ableitung des Beispielsatzes *Kamera1 speicher die Einstellung als Shot1*.

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

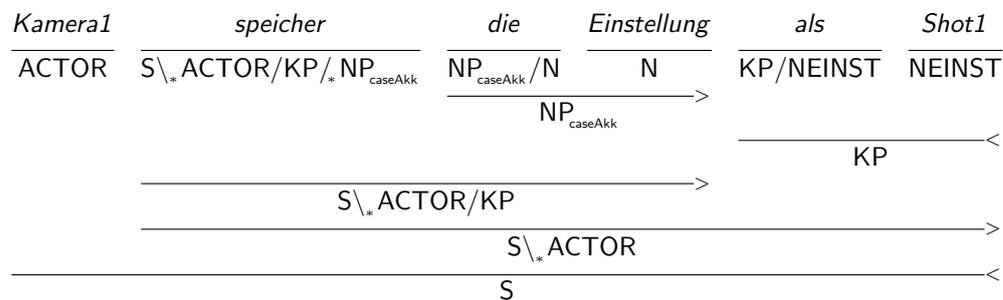


Abbildung 5.41: Ableitung eines Speichern-Kommandos mit Hilfe der CCG.

Speichern legt fest, dass es vorangestellt einen ACTOR und nachgestellt sowohl eine Nominalphrase NP im Akkusativ als auch eine Konjunkionalphrase KP benötigt. Für den Beispielsatz ergibt die Kombination aus Artikel (*die*) und Nomen (*Einstellung*) die nötige Nominalphrase NP und die Konjunktion *als* gefolgt von dem Substantiv *Shot1* die Konjunkionalphrase. Abbildung 5.42 zeigt die logische Form eines Speichern-Kommandos.

```
@w1:action(save ^
  <VERB>speicher ^
  <ACTOR>(w0:actor ^ Kamera1 ^
    <num>sg-X ^
    <gender>genderFem-X) ^
  <TARGET>(x1 ^
    <konj>als ^
    <target>(w5:einstellung ^ Shot1 ^
      <num>sg-X ^
      <gender>genderMasc-X)) ^
  <WHAT>(w3:obj ^ Einstellung ^
    <num>sg-X ^
    <gender>genderFem-X ^
    <art>article))
```

Abbildung 5.42: Logische Form eines Speichern-Kommandos.

Für die logische Ableitung des Beispielsatzes wird die Kategorie von *speichern* durch den logischen Ausdruck $\models @_{action}(save \wedge VERB_V \wedge ACTOR_X \wedge TARGET_T \wedge WHAT_Y)$ ergänzt. Das Nominal WHAT ergibt sich aus der logischen Ergänzung $\models @(art_X)$ der Kategorie NP / N und aus der logischen Ergänzung $\models_{obj}(X)$ der Kategorie N. Die Kategorie von *als* KP / NEINST wird

durch $\models (target_X \wedge konj_Y)$ und die Kategorie NEINST von *Shot1* wird durch $\models_{Einstellung} (E)$ ergänzt. Diese beiden Formen beschreiben das Nominal TARGET.

5.6 Anfahrt einer bestimmten Einstellung

Die einfachste Form der Kamerasteuerung ist die Nennung der gewünschten Einstellungsgröße. Sie bezieht sich einerseits auf das Anfahren einer Standard-Einstellung und andererseits auf das Laden und Anfahren einer zuvor gespeicherten Position. Alle Kommandos mit Bezug auf das Anfahren einer bestimmten Einstellung werden im Folgenden als GotoShot-Kommandos bezeichnet. Abbildung 5.43 zeigt einige Beispiele zur Anfahrt einer bestimmten Einstellung.

```
Kamera1 zeige mir eine Halbnahe
Kamera1 gib mir eine Nahe
Kamera1 gehe auf eine Totale
Kamera1 zeige mir Shot1
```

Abbildung 5.43: Beispiele zur Anfahrt einer Einstellung.

Die Beispiele zeigen, dass das Anfahren einer bestimmten Einstellung alternativ durch die Indikativ-Form der Verben *zeigen*, *geben* oder *gehen* gefordert werden kann. Abbildung 5.44 zeigt die Struktur der Kommandos.

```
[GOTOSHOT-CMD] := [ACTOR] [zeigen] [OBJ_DAT] [NP_EINST]
[GOTOSHOT-CMD] := [ACTOR] [gehen|fahren] [PP_EINST]
```

Abbildung 5.44: Struktur von Kommandos zur Anfahrt einer Einstellung.

Sätze mit *zeigen* enthalten ein Objekt im Dativ, das angibt, wem die jeweilige Einstellung gezeigt werden soll. Im Normalfall geht die Anweisung vom Regisseur aus, deshalb repräsentiert dieses Objekt meist das Pronomen *mir*. Dem Pronomen folgt eine Nominalphrase NP_EINST, die für eine Einstellung steht. Wird ein Satz mit den Verben *fahren* bzw. *gehen* gebildet, folgt eine Präpositionalphrase PP_EINST. Diese wird durch die Präposition *auf* eingeleitet und durch eine Nominalphrase im Akkusativ bzw. durch eine Einstellungsbezeichnung ergänzt. Abbildung 5.45 zeigt oben die Ableitung von *Kamera1 zeige mir eine Totale* und *Kamera1 gehe auf eine Nahe*.

Diese zwei Sätze sind häufig gebrauchte Anweisungen, um eine Kamera auf eine bestimmte Einstellung hin ausrichten zu lassen. *Zeige* ist einer Kategorie zugeordnet, die führend einen ACTOR und nachfolgend zwei Nominalphrasen NP benötigt. Die erste NP steht dabei im Dativ,

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

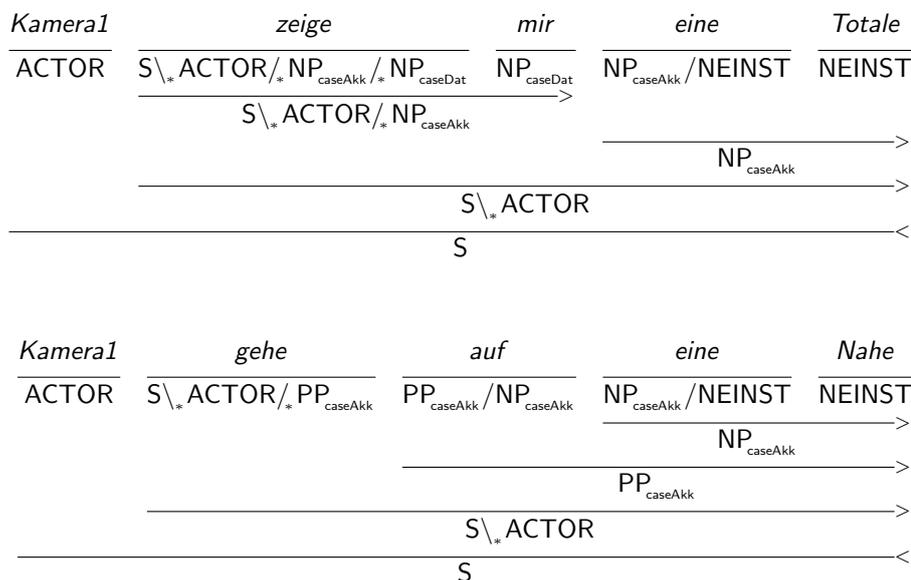


Abbildung 5.45: Ableitung eines GotoShot-Kommandos.

die zweite im Akkusativ¹. Die Zuordnung der Präposition *mir* zur Kategorie NP stellt einen Spezialfall dar (vgl. Kapitel 5.3.1). Dieser erlaubt es, den allgemeinen Fall des Kommandos auch für die Präposition *mir* zu verwenden, obwohl eine Präposition keine Phrase darstellt. Würde das hier behandelte Kommando von einem Dritten abgegeben werden, müsste die Formulierung nach *Kamera1 zeige dem Regisseur eine Nahe* abgeändert werden. Hier ist die Nominalphrase klar als solche zu erkennen. Um keine weitere Kategorie für das Verb *zeige* anlegen zu müssen, wurde *mir* ebenfalls der Kategorie NP zugeordnet.

In Abbildung 5.45 unten wird die Ableitung des GotoShot-Kommandos bei Verwendung des Verbs *gehe* gezeigt. Hier geht dem Verb ebenfalls ein ACTOR voraus. Eine Präpositionalphrase PP folgt dem Verb. Sie wird durch die Präposition *auf* und der Nominalphrase, die durch *eine* und *Nahe* entsteht, konstruiert. Die Präpositionalphrase muss ein Argument im Akkusativ besitzen. Deshalb wird der Kasus durch die einzelnen Kategorien weitergereicht. Abbildung 5.46 zeigt die logische Form des GotoShot-Kommandos aus Abbildung 5.45 unten.

Die Kategorie des Verbs *gehen* wird dabei durch die Anweisung $\models @_{\text{action}}(\text{gotoShot} \wedge \text{VERB}_Y \wedge \text{ACTOR}_X \wedge \text{PATIENT}_Z)$ ergänzt. Sie beschreibt eine *gotoShot*-Aktion mit den Nominalen *VERB*, *ACTOR* und *PATIENT*. Das Verb selbst trägt bei diesem Ausdruck keine Information und wird nur der Vollständigkeit halber angegeben. Unter *ACTOR* ist die jeweilig

¹Die OpenCCG ist rechtsbindend. Deshalb steht die Akkusativ-NP-Kategorie nach der Dativ-NP-Kategorie.

```

@w1:action(gotoShot ^
  <VERB>gehe ^
  <ACTOR>(w0:actor ^ Kamera1 ^
    <num>sg-X ^
    <gender>genderFem-X) ^
  <PATIENT>(x1:Einstellung ^
    <art>eine ^
    <prep>auf ^
    <shot>(w4:einstellung ^ Nahe ^
      <num>sg-X ^
      <gender>genderFem-X)))

```

Abbildung 5.46: Logische Form eines GotoShot-Kommandos.

angesprochene Kamera angegeben. *PATIENT* beschreibt dabei die Einstellung, die eingenommen werden soll. In dem Beispiel wird sie durch die Präpositionalphrase *auf eine Nahe* beschrieben. *PATIENT* gibt dabei die Präposition an und *eine Nahe* die gewünschte Einstellung, definiert durch eine Nominalphrase.

Für das Kommando *Kamera1 zeige mir Shot1* muss der Kategorie von *zeige* die logische Form $\models @_{action}(gotoShot \wedge VERB_V \wedge ACTOR_X \wedge VERB_Y \wedge PATIENT_P \wedge REFP_R)$ hinzugefügt werden. Sie ergänzt den bisherigen Ausdruck durch die Angabe des Reflexivs *REFP_R*.

5.7 Zeitwährende Kommandos

Die bisher vorgestellten Kommandos sind gezielt für die Steuerung und Korrektur eines Kamerasystems verwendbar. Die Steuerung fällt nach Ausführung des Kommandos in einen stabilen Zustand zurück und wartet auf weitere Anweisungen. Für die *Einstellungsfindung* sind die bisher vorgestellten Kommandos gut zu verwenden. Allerdings zwingen sie den Benutzer stets zu einer Interaktion, sobald sich Elemente des Sets (meist der Moderator) und damit das Fernsehbild verändern. Eine intelligente Kamerasteuerung muss in der Lage sein, auf Änderungen im Bild zu reagieren. Aus diesem Grund wurden *zeitwährende Kommandos* integriert, die das System so lange berücksichtigt, bis sie durch ein Stopp-Kommando aufgehoben werden. Abbildung 5.47 zeigt einige Beispiele für Kommandos, die über einen längeren Zeitraum Gültigkeit behalten.

Für die anhaltende Wirkung eines Kommandos werden die Verben *setze* und *zeige* durch das Verb *halte* ersetzt. Die grundsätzliche Struktur der jeweiligen Kommandos ändert sich dabei nicht. Sie ist, abgesehen vom Verb, identisch zu der in Abbildung 5.27 bzw. 5.44. Auf eine

5. AUTOMATISCHE VERARBEITUNG VON NATÜRLICHER SPRACHE IM STUDIO

Kamera1 halte den Moderator mittig
 Kamera1 halte den Moderator auf der linken Seite
 Kamera1 halte den Moderator auf der rechten Seite
 Kamera1 halte eine Halbnah
 Kamera1 halte mir eine Totale

Abbildung 5.47: Beispiele für zeitwährende Kommandos.

detaillierte Präsentation wird an dieser Stelle verzichtet. Sie ergibt sich aus bereits vorgestellten Kommandos. Demnach geht dem Verb *halte* immer der Adressat *ACTOR* voraus. Es folgen je nach Intention ein Objekt *OBJ_AKK* im Akkusativ und eine Eigenschaft *PROP*, eine Einstellung *EINST* oder ein Reflexivpronomen *REF* gefolgt von einer Einstellung *EINST*. Auf die Ableitung eines Beispielsatzes wird hier ebenfalls verzichtet. Siehe dazu Abbildung 5.28 bzw. 5.45. Die logische Form für das Beispiel *Kamera1 halte den Moderator mittig* ist in Abbildung 5.48 dargestellt.

```
@w1:action(composeObjectAdj ^
  <VERB>halte ^
  <ACTOR>(w0:actor ^ Kamera1 ^
    <num>sg-X ^
    <gender>genderFem-X) ^
  <PATIENT>(w3:obj ^ Moderator ^
    <num>sg-X ^
    <gender>genderMasc-X ^
    <art>den) ^
  <POSITION>(w4:prop ^ mittig))
```

Abbildung 5.48: Logische Form eines zeitwährenden Kommandos.

Die Kategorie des Verbs *halte* wird um den logischen Ausdruck $\models @_{action}(composeObjectAdj \wedge VERB_X \wedge ACTOR_Y \wedge PATIENT_Z \wedge POSITION_P)$ ergänzt. In der logischen Aussage ändert sich ausschließlich das Verb zu *halte*.

Kapitel 6

Das Aktionsmodell zur Handlungsgenerierung

Der Lebenszyklus einer Sendung beginnt für einen Kameramann mit der *Einstellungsfindung*. Hier werden die Grundlagen für die spätere Sendung gelegt. Aus der Grafik-Abteilung und von der Redaktion existieren Vorgaben, die grob festlegen, wie die Sendung später zu gestalten ist. Diese Vorgaben basieren auf Screenshots, die Einstellungen der späteren Sendung zeigen. Sie sind oft skizzenhaft zusammen getragen und können im Studio nicht detailgetreu nachgestellt werden. Es ist Aufgabe der *Einstellungsfindung* Einstellungen so abzustimmen, dass sie als verbindliche Vorlage für die jeweilige Sendung verwendet werden können. In der vorliegenden Arbeit wird gerade dieser Prozess als essenziell angesehen, weil sich hier bereits Informationen sammeln lassen, die für die spätere Sendung wichtig werden könnten. Der Schwerpunkt der Ausführungen in dieser Arbeit liegt in der Unterstützung der *Einstellungsfindung*. Vor allem in dieser Phase werden unterschiedliche Joystick-Bewegungsformen verwendet. Ergänzt werden die damit verbundenen Kommandos durch Anweisungen zum Laden und Speichern von Einstellungen sowie durch eine automatische Anpassung der Einstellung an unterschiedliche Moderatorengrößen (Kompensationsstrategie – vgl. Kapitel 4.3.2). Im vorhergehenden Kapitel wurde die Verarbeitung der natürlichen Sprache in eine logische Struktur behandelt. In diesem Kapitel wird die Weiterverarbeitung dieser Struktur bis hin zur Bewegungsgenerierung vorgestellt.

Ein intelligentes System muss auf unterschiedliche Einflüsse reagieren können. Für die Kamerasteuerung liegen die Einflussquellen insbesondere im Domänenwissen (Knowledge), in der Weltinformation (World-Information), in der Bildinformation (Image-Knowledge) und in der

6. DAS AKTIONSMODELL ZUR HANDLUNGSGENERIERUNG

adäquaten Verarbeitung von externen Eingaben (external Input). Abbildung 6.1 zeigt die Einflüsse, projiziert auf eine intelligente Kamerasteuerung.

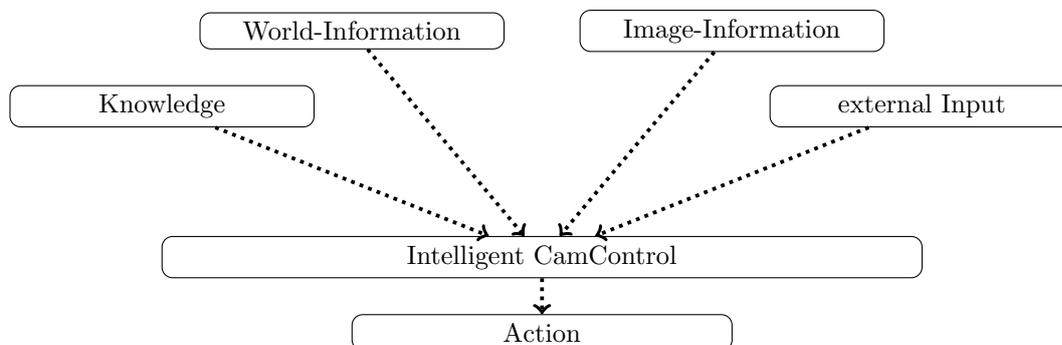


Abbildung 6.1: Modell zur intelligenten Kamerasteuerung. Aus unterschiedlichen Einflüssen muss die intelligente Steuerung eines Kamerasystems eine Aktion generieren.

Unter Domänenwissen werden alle Informationen aus dem Bereich der Kameraführung sowie dem Umfeld des Studios und des zu steuernden Gerätes zusammengefasst. Darunter fällt explizit das Wissen, was unter einer bestimmten Einstellung zu verstehen ist und wie man diese einnehmen kann sowie Informationen über Grenzen der Beweglichkeit der verwendeten Gerätschaft. Die individuelle Vorstellung zur Ästhetik fällt ebenso unter diesen Bereich wie das Verstehen eines gesprochenen Satzes. Das System muss wissen, welche Aktion mit dem angesprochenen Kommando gemeint ist. Wissen ist im hier verwendeten Sinn all diejenige Information, die unabhängig vom Zustand der Welt ist.

Die Weltinformation enthält die Informationen über das Set und die Kamera. Hier ist definiert, welche Elemente zum Set gehören und wo sich diese in der betrachteten Welt befinden. Ferner gehören die Positionsinformationen (Pose) der Kamera zu dieser Art von Information.

Die Bildinformation enthält Positionen und Größen der einzelnen Objekte im Bild. Diese können anhand von unterschiedlichen Kameramodellen aus der Weltinformation extrahiert werden. Allerdings bleibt dieser Schritt im Weiteren unberücksichtigt. In der vorliegenden Arbeit wird versucht, mit den Informationen, die einem menschlichen Kameramann zur Verfügung stehen, intelligent eine Kamera zu steuern. Im Fokus steht dabei die visuelle Information des Bildes.

Durch die externen Eingaben erfährt das System was es als nächstes tun soll. In dem hier beschriebenen System erfolgt dieser Vorgang durch die Sprachverarbeitung und damit im Wesentlichen über den Einfluss *Wissen* (Knowledge). In einem realen Szenario ist es allerdings denkbar, dass ein System z.B. auch durch ein Ablaufsystem auf Basis von XML angesteuert

wird. Die *Intelligenz* zur Verarbeitung natürlicher Sprache ist dann nicht mehr erforderlich und der Einfluss externer Eingaben gewinnt damit wieder an Eigenständigkeit. Externe Eingaben sind alle nicht-verbale Eingaben an das System.

Aus den genannten Eingaben generiert eine intelligente Steuerung eine gewünschte Bewegung der Kamera. Diese Bewegung sollte der eines menschlichen Kameramanns entsprechen.

6.1 Der Regelkreis

Das System durchläuft kontinuierlich drei Zustände: *Percept*, *Reflect* und *Act*. Es folgt damit einer Vorgehensweise, die der menschlichen Problemlösung entspricht: Wahrnehmen und Erkennen, Nachdenken, als Abgleich der aktuellen Situation mit dem Ziel, und schließlich die Ausführung von angemessenen Maßnahmen. Abbildung 6.2 zeigt diesen Fortgang durch die einzelnen Zustände.

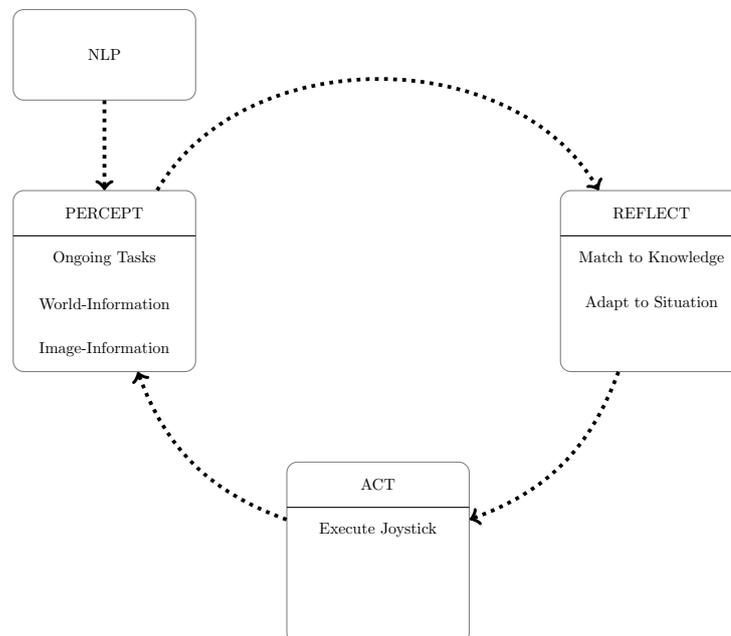


Abbildung 6.2: Regelkreis zur Kameraführung. Auf das Wahrnehmen der Situation (Percept) folgt das Folgern (Reflect) und Ausführen der notwendigen Schritte (Act). Die Sprachverarbeitung (natural language processing – NLP) wandelt verbale Eingaben in eine logische Form und stellt sie dem Regelkreis zur Verfügung.

6.1.1 Percept

Unter *Percept* fällt die komplette Wahrnehmung der Welt. Dazu gehören Aktionen, die bereits in Ausführung sind (aktuelle Aktionen), das Wahrnehmen von Nutzer-Eingaben sowie Änderungen an der Welt bzw. im Bild. Die Eingabe der logischen Fragmente eines Sprachbefehls fällt ebenso in diesen Zustand wie die Änderungen an der Kamera-Position oder am Set.

6.1.2 Reflect

Ist die Phase *Percept* abgeschlossen, wird in den Zustand *Reflect* gewechselt. Hier wird ein Abgleich der neu gewonnenen Information mit der Wissensbasis durchgeführt und dabei die Wissensbasis erweitert. In hierarchischer Reihenfolge werden neue Nutzereingaben mit den aktuell vorhandenen abgeglichen und aus den verbal eingegebenen logischen Fragmenten neue Ziele definiert. Die Ziele hängen von den vorher getroffenen Grundeinstellungen ab. Sind die Ziele unmittelbar erreichbar, können sie im letzten Schritt der Handlungshierarchie direkt ausgeführt werden. Ist die unmittelbare Ausführung nicht möglich, muss das System die Ziele abändern und Alternativ-Ziele einführen.

6.1.3 Act

Aus den gewonnenen Informationen wird schließlich im Zustand *Act* die gewünschte Bewegungsinformation an das Kamerasystem weitergeleitet. Die dadurch entstehenden Änderungen an der Welt werden im nächsten Durchlauf des Zyklusses verarbeitet.

Abbildung 6.2 zeigt den Regelkreis schematisch. In dieser Darstellung ist die Sprachverarbeitung (NLP) als eigene Entität dargestellt. Das Wissen, das sich hinter der Sprachverarbeitung verbirgt, zählt zwar zur Intelligenz des Systems, aber nicht zur Intelligenz, die für die Aktionsauswahl zuständig ist. Die verbalen Nutzereingaben werden in logische Fragmente zerlegt, die dann der Wissensbasis zur Verfügung gestellt werden. Anhand dieser Fragmente wird daraus eine Bewegungshypothese abgeleitet, die unter Berücksichtigung des bekannten Wissens (z.B. der physikalischen Grenzen) angepasst und schließlich ausgeführt wird.

6.2 Reflect – eine hierarchische Regelstruktur

In der Domäne eines Fernsehstudios ist das Verhalten der Kameraleute stark durch Regeln geprägt. Ausgehend vom Ausbildungswissen, über eigene Erfahrungen bis hin zu Sender- und Sendungseigenheiten werden Einstellungen anhand von Regeln definiert. Für den Kameramann

6.2 Reflect – eine hierarchische Regelstruktur

ist das Storybook, in dem die Einstellungen im Einzelnen beschrieben sind, der Ausgangspunkt für die Regeln. Für die jeweilige Sendung wird diese Grundlage evtl. durch Eigenheiten des aktuellen Regisseurs oder persönliche Auslegung der Vorlage ergänzt. Eine Überführung dieser Vorgaben in eine regelbasierte Wissensbasis ist naheliegend.

Die Regeln sind in unterschiedliche, hierarchische Schichten eingeteilt. So wird sichergestellt, dass die Abarbeitung der Regeln einer bestimmten Reihenfolge genügt und damit eine strukturell ähnliche Behandlung für unterschiedlichste Eingaben möglich ist. Ein zentrales Beispiel ist der Abgleich der gewünschten Kamerabewegung mit dem zur Verfügung stehenden Arbeitsraum. Wird eine Bewegung im Grenzbereich des Arbeitsraums nach außen hin gefordert, muss sie rechtzeitig gestoppt und im Idealfall durch eine optisch ähnlich erscheinende Bewegung ersetzt werden. In der vorliegenden Arbeit werden sechs Hierarchiestufen unterschieden, die zyklisch nacheinander durchlaufen werden. Abbildung 6.3 zeigt diese Hierarchie.

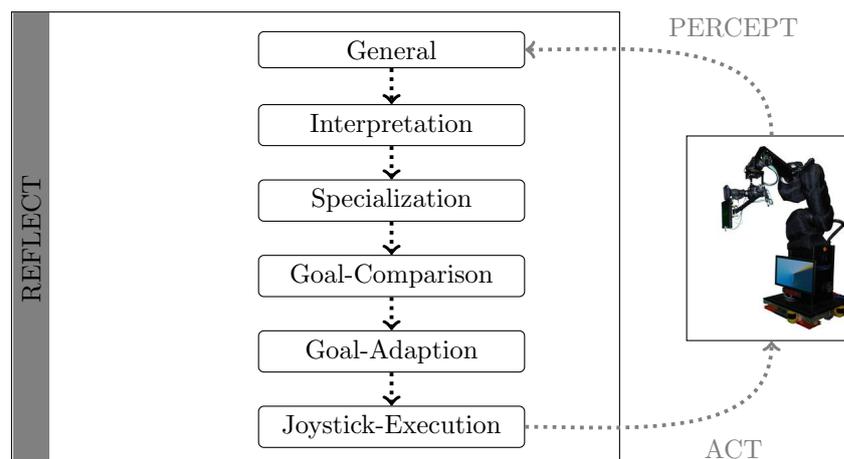


Abbildung 6.3: Hierarchie-Stufen im *Reflect*-Zustand. Die *General*-Schicht stellt sicher, dass alle nötigen Informationen verfügbar sind. Die *Interpretation*-Schicht übernimmt die von der Sprachverarbeitung generierten Fragmente in die Wissensbasis. Hier wird durch die *Specialization*-Schicht evtl. fehlende Informationen zu diesen Fragmenten hinzugefügt. In der *Goal-Comparison*-Schicht wird überprüft, ob das geforderte Ziel bereits erreicht ist. Die *Goal-Adaption*-Schicht erlaubt es unerreichbare Ziele durch ähnliche Aktionen zu ändern und die *Joystick-Execution*-Schicht übergibt die Bewegungsinformation an das Kamerasystem.

Ein weiterer Aspekt der Hierarchisierung ist die bessere Wartbarkeit des Systems. Müssen Regeln geändert werden, so ist klar, wie diese abgearbeitet werden und welche Regeln auf welche Regeln folgen. Nach einer Änderung müssen so nur die nachfolgenden Hierarchiestufen für die Adaption berücksichtigt werden.

6.2.1 General

Die *General*-Schicht ist der Startpunkt eines jeden Durchlaufs der *Reflect*-Struktur. In diesem Strukturelement werden alle Informationen erfasst und berücksichtigt, die allgemein für das System gelten und nicht direkt für die Fahrt-Generierung relevant sind. Hier werden Warnungen generiert, wenn nötige Schnittstellen nicht gesetzt sind bzw. sich das System nicht in einem fahrbereiten Zustand befindet. Dies gilt insbesondere, wenn keine Informationen von der Welt bzw. vom Kamerasystem vorliegen. Die Informationen über die Welt werden in Form von Prädikaten (Facts) formuliert. Die Prädikate enthalten z.B. Informationen über die Position einzelner sowie gruppierter Set-Elemente (Gruppenprädikate) oder geben studiotypische Größen, wie z.B. die *Luft über dem Kopf* oder die Augenhöhe des Moderators *eyeHeight* an. Außerdem werden Kamerasystemzustände, wie *joystickRunning*, an das Kontroll-System übergeben. Für die Dimensionen der Objekte werden die Prädikate *width2d*, *height2d* sowie *size2d* verwendet. *width2d* gibt die Breite eines Objekts, *height2d* die Höhe des Objekts im Bild an. *size2d* gibt das jeweilige Maximum von *width2d* und *height2d* zurück.

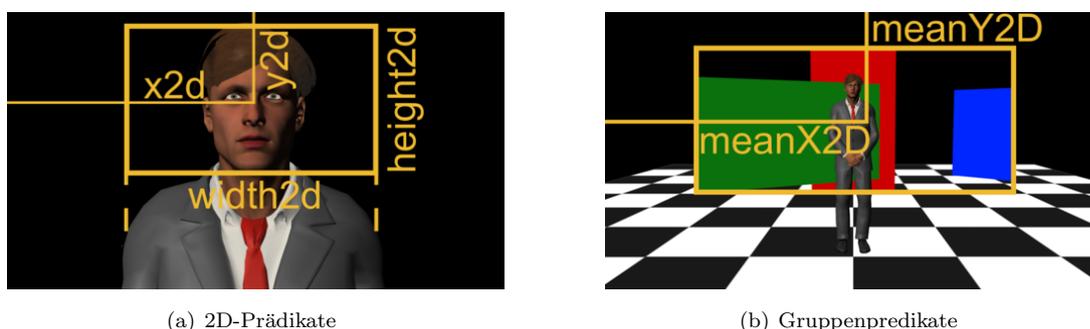


Abbildung 6.4: Repräsentation der 2D-Prädikate. $x2d/y2d$ beschreibt die Position eines Objekts im Bild in Abhängigkeit vom linken/oberen Bildrand. $height2d/width2d$ beschreibt die Höhe/Breite eines Objekts prozentual in Abhängigkeit der Bildbreite. $meanX2D/meanY2D$ beschreibt die Position einer Gruppe von Objekten im Bild so wie $x2d/y2d$ für ein Objekt.

Abbildung 6.4 zeigt die Definition der 2D- sowie der Gruppen-Prädikate. Alle Objekte werden durch Rechtecke ausgedrückt. Diese befinden sich auf der Bildebene um das jeweilige Objekt. Die Position des Objekts wird relativ zur linken oberen Ecke prozentual jeweils zur Breite bzw. zur Höhe des Bildes festgelegt. $x2d$ beschreibt den Abstand vom linken Bildrand, $y2d$ den Abstand vom oberen Bildrand. Das Objekt *Moderator* wird durch ein Rechteck modelliert, das um den Kopf des Moderators verläuft. Der Kopf des Moderators gibt die Höhe (*height2d*), der Abstand der Schultern die Breite (*width2d*) des Rechtecks vor.

6.2.2 Interpretation

In dieser Ebene werden die logischen Strukturen der Sprachverarbeitung auf die internen Datenstrukturen des Systems abgebildet. Es werden die logischen Ausdrücke aus der Sprachverarbeitung in die tatsächlichen Aktionen überführt und dann weiterverarbeitet. Diese Schicht trennt die Eingabe von der Verarbeitung, sodass das System schnell auf beliebige Eingabesysteme adaptiert werden kann.

6.2.3 Specialization

Diese Schicht verfeinert einige der in der *Interpretation*-Schicht aufgenommenen Kommandos. Dies kommt insbesondere dann zum Tragen, wenn die Kommandos allgemeingültige Begriffe, wie z.B. *auf die linke Seite* oder *Halbnahe* enthalten. Die Informationen zur Verfeinerung dieser Begriffe werden in dieser Schicht vorgehalten. Hier wird auch definiert, was z.B. unter einer *Halbnahen-Einstellung* in der aktuellen Situation tatsächlich zu verstehen ist. In den Kapiteln 6.3 bis 6.11 werden die Funktionsweise dieser und der *Interpretation*-Schicht nach Kommando unterschieden vorgestellt.

6.2.4 Goal-Comparison

In dieser Schicht wird überprüft, ob der erreichte Zustand ausreicht, um die aktuelle Aktion als erfüllt anzusehen. Der Schritt bezieht sich auf die Entscheidung des vorhergehenden Durchlaufs. Hier wird dafür gesorgt, dass erfüllte Aktionen aus der Verarbeitungskette entfernt werden.

PseudoCode 1 Goal-Comparison

```
1: for all JoystickActions a do  
2:   if a.hasReached == true then  
3:     context.saveAction(a);  
4:     remove(a);  
5:   end if  
6: end for
```

PseudoCode 1 gibt Aufschluss über die Funktionsweise dieser Schicht. Ergibt die Auswertung der Aktion, dass die angestrebte Position erreicht ist, wird die Aktion im Kontext gespeichert und daraufhin aus der Wissensbasis entfernt. Durch das Aufnehmen der ausgeführten Aktionen in den Kontext, kann bei einem folgendem *zurück*- oder *weiter*-Kommando die Historie der Kommandos durchlaufen und so die letzte Aktion wiederholt oder invertiert werden.

6.2.5 Goal-Adaption

Erreicht eine der bewegten Achsen ein technisches Limit, kann die Aktion nicht zu Ende ausgeführt werden. In der *Goal-Adaption*-Schicht werden insbesondere die Limits des Systems berücksichtigt und Fahr-Kommandos abgeändert oder gegebenenfalls abgebrochen. Das Wechseln einer Bewegung entlang der Längsachse der Kamera auf eine Bewegung des Zooms ist ein Beispiel für eine solche Adaption. Außerdem kann für Kamerasysteme mit sphärischem Arbeitsraum bei angeforderter Achse *Radius* diese selbstständig auf die Achse Z geändert werden, damit der Radius des Systems vergrößert werden kann.

PseudoCode 2 Goal-Adaption

```
1: for all JoystickActions a do
2:   if inLimit(a.axis)  $\wedge$  a.direction towards Limit then
3:     if not findAlternative() then
4:       remove(a);
5:     else
6:       a.updateAxis()
7:     end if
8:   end if
9: end for
```

PseudoCode 2 zeigt den schematischen Ablauf in dieser Schicht. Für alle Joystick-Aktionen wird, wenn sich die beteiligte Achse am Limit befindet und die Kamera sich weiter in Richtung des Limits bewegen soll, eine Alternative gesucht. Kann keine gültige Alternative gefunden werden, wird die Aktion aus der Wissensbasis entfernt.

6.2.6 Joystick-Execution

Die in dieser Arbeit vorgestellte, intelligente Kamerakontrolle verwendet ein typisches Interface für fernsteuerbare Studio-Systeme. Die Steuerung geschieht über die üblichen Achsen wie Pan, Tilt, Roll, X bzw. Angle (=radiale Bewegung der Kamera um die Basis), Y bzw. Radius (Entfernung der Kamera zur Basis) und Height. In der *Joystick-Execution*-Schicht werden die berechneten Fahrgeschwindigkeiten auf das Ausgabe-Interface gelegt und so die tatsächliche Aktion ausgelöst. Das Interface erlaubt dabei die gleichzeitige Ausführung unterschiedlicher Fahrbefehle, sofern sich diese nicht wechselseitig ausschließen. PseudoCode 3 zeigt die Funktion dieser Schicht.

PseudoCode 3 JoystickExecution

```

1: SpeedMap map;
2: for all JoystickActions a do
3:   if a.hasReached() == true AND NOT a.nodelete then
4:     remove a;
5:   else
6:     map.update(a.calcSpeed());
7:   end if
8: end for
9: send(map);

```

Für jede im System vorhandene Aktion wird überprüft, ob sie bereits erfüllt (*hasReached()*) wurde. Ist dies der Fall und ist für die Aktion das Flag `nodelete` (vgl. Kapitel 6.11) nicht gesetzt, wird sie aus dem System entfernt. Ist dies nicht der Fall wird die Geschwindigkeit für die, mit der Aktion verknüpften, Achse berechnet und an das Kamerasystem gesendet.

In den folgenden Kapiteln werden die einzelnen vom System unterstützten Kommandos diskutiert. Sie sind nach aufsteigender Komplexität jeweils für die Phasen *Einstellungsfindung* und *Sendekontrolle* aufgeführt.

6.3 Stopp-Kommando

Das Stopp-Kommando zählt auf Grund seiner Struktur zu den einfachsten und gleichzeitig wichtigsten Kommandos des Steuerungs-Systems. Das System stoppt alle Fahraktivitäten und geht in einen stabilen Zustand über. Es erhält innerhalb der *Interpretation*-Schicht die höchste Priorität und wird deshalb an erster Stelle in der Bearbeitung berücksichtigt.

Die Verarbeitung dieser Anweisung geschieht in drei Schritten, die untereinander wieder eine hierarchische Folge bilden. Nach Eingabe des Stopp-Kommandos werden alle Aktionen aus der Wissensbasis entfernt. Daraufhin folgen die logischen Strukturen, die neben der Stopp-Anweisung noch im System existieren könnten und schließlich wird die Anweisung selbst entfernt. Das System ist dann wieder in einen Zustand gebracht, in dem es neue Anweisungen entgegen nehmen kann. Abbildung 6.5 zeigt den hierarchisch strukturierten Ablauf grafisch.

Der logische Ausdruck eines Stopp-Kommandos wird durch die in Kapitel 5 vorgestellte Grammatik in zwei Fragmenten `Stop:ACTOR` und `Stop:STOP` an das System übergeben. `Stop:ACTOR` enthält die Information, welche Kamera gestoppt werden soll und `Stop:STOP` die Stopp-Information selbst. Vor dem Doppelpunkt wird jeweils der Anweisungstyp der übergebenen

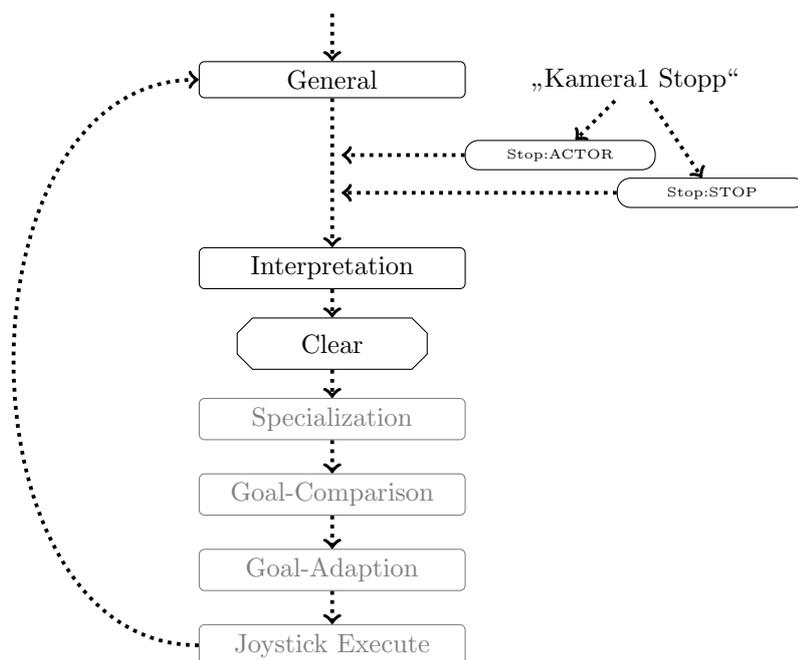


Abbildung 6.5: Schematischer Ablauf eines Stopp-Kommandos. Die grau dargestellten Zustände werden bei diesem Kommando nicht durchlaufen.

Fragmente dargestellt. Im hier vorgestellten Beispiel ist der Typ *Stop*. Der Typ der Fragmente wird durch die Sprachverarbeitung festgelegt und erleichtert die Identifikation der Kommandos.

Die Stopp-Fragmente entziehen den nachgeordneten Ebenen die Aktionen. Sie sind deshalb in Abbildung 6.5 grau dargestellt. Das System wird in diesem Durchgang stoppen und alle vorherigen Aktionen einstellen. Wird kein *ACTOR* angegeben, so stoppt das System ebenfalls. Sind mehrere Systeme in einem Studio im Einsatz stoppen so alle Systeme.

6.4 Joystick-Kommandos

Unter Joystick-Kommandos werden hier alle Anweisungen verstanden, die direkt eine Achse des Kamerasystems ansprechen. Die logische Form dieser Kommandos gliedert sich in vier Fragmente: Den Adressat (*ACTOR*), die Richtung (*DIR*), das Verb (*VERB*) und optional eine Quantität (*QUANTITY*). Abbildung 6.6 zeigt den schematischen Ablauf für *Kamera1 fahre ein bisschen höher* innerhalb des Reflect-Zustandes.

Das Fragment *ACTOR* bezeichnet, wie in allen anderen Kommandos, den Adressaten. Das Fragment *DIR* gibt zusammen mit dem *VERB* die Richtung der Bewegung an (vgl. Kapitel 5.2).

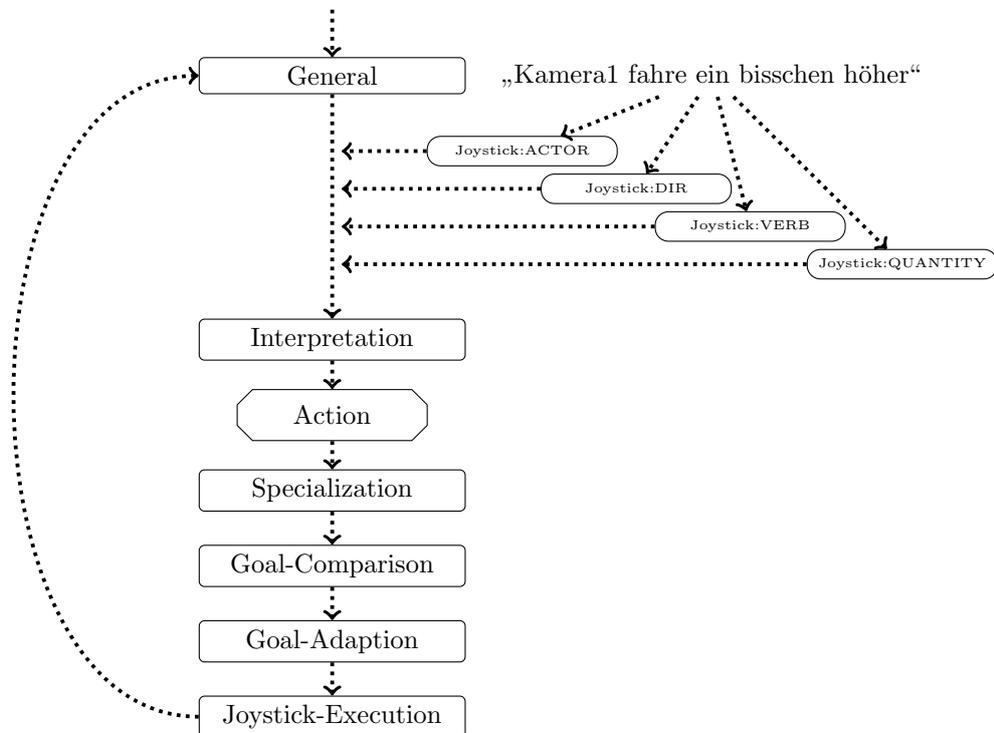


Abbildung 6.6: Schematischer Ablauf einer Joystick-Anweisung. Bei einer Joystick-Anweisung werden die logischen Fragmente in der *Interpretation*-Schicht in die Wissensbasis aufgenommen und hier zu einer *Action* transformiert. In der *Goal-Comparison*-Schicht wird überprüft, ob die *Action* bereits als erfüllt angesehen werden kann. Falls nicht, wird in der *Goal-Adaption*-Schicht bei Unerreichbarkeit des Ziels dieses angepasst. In der *Joystick-Execution*-Schicht werden die Fahrinformationen an das Kamerasystem weitergereicht.

Neben einer Änderung der Richtung bzw. der Orientierung kann auch eine Modifikation der Brennweite (Zoom) angefordert werden. Der Ausdruck für die Quantität (QUANTITY) beschreibt qualitativ, wie weit verfahren werden soll. In der *Interpretation*-Schicht wird aus den logischen Fragmenten eine Aktion generiert, die die übergebenen Informationen speichert.

Anhand der gespeicherten Werte wird in der *Goal-Comparison*-Schicht getestet, ob die vorgegebene Bewegung bereits vollständig ausgeführt wurde. Im positiven Fall wird die Aktion (*Action*) aus der Wissensbasis entfernt. Ist dies nicht der Fall, wird in der *Goal-Adaption*-Schicht die Fahrbewegung auf Durchführbarkeit getestet und evtl. geändert. Am tiefsten Punkt der Bearbeitung (in der *Joystick-Execution*-Schicht) werden die Geschwindigkeiten der einzelnen Fahrbefehle berechnet und an das Kamerasystem übermittelt. Die Geschwindigkeit ist abhängig von der 2D-Entfernung des jeweils fokussierten Objekts zum Ziel der Fahrt. Für die

Berechnung der Geschwindigkeit wird die Basisgeschwindigkeit der jeweiligen Achse mit dem Abstand in Bild-Prozent-Punkten multipliziert (6.1).

$$v_{axis} = v_{base} * dist_{cur \rightarrow goal} \quad (6.1)$$

6.4.1 Fahrdistanz

Die Bestimmung der Fahrdistanz wird, sofern nicht durch das Kommando explizit festgelegt, durch das System definiert. Für alle Achsen wird die Änderung der Werte durch die Verschiebung bzw. Skalierung der Objekte im Bild gemessen. Für die Quantität **VIEL** wird eine Positionsänderung um ca. 20%, für **MITTEL** um ca. 10% und für **WENIG** um ca. 5% durchgeführt (6.2).

$$d_{axis} = \begin{cases} 20\% & \text{falls Quantität} == \text{VIEL} \\ 10\% & \text{falls Quantität} == \text{MITTEL} \\ 5\% & \text{falls Quantität} == \text{WENIG} \end{cases} \quad (6.2)$$

Für die Achsen Pan und Y ist die horizontale Bewegung der Objekte im Bild, bei Tilt bzw. Z die vertikale und für Zoom bzw. X jeweils die Breite des Moderators auf dem Bild relevant. Die hier angegebenen Werte wurden empirisch ermittelt. Sie können jedoch von jedem Benutzer individuell eingestellt werden.

Alternativ zu diesem Ansatz wurde ein gerätebasierter Ansatz getestet. Dieser verfährt die ausgewählte Achse um einen bestimmten Wert, abhängig von der geforderten Quantität. Die Veränderungen im Bild werden nicht berücksichtigt. Für die kartesischen Achsen X, Y und Z wird die Kamera abhängig von der Quantität **VIEL**, **MITTEL** und **WENIG** jeweils um 15cm, 10cm und 5cm bewegt. Für die Orientierungsachsen werden die Änderungen in Grad (3°, 5° und 7°), bei der Zoom-Achse in Prozent (4%, 6% und 10%) ausgeführt. In Tests hat sich gezeigt, dass der Ansatz in allgemeiner Verwendung zu unbefriedigenden Ergebnissen führt. Bei einem aktiven zeitwährenden Kommando wird diese Methode dennoch angewandt, weil sie hilft gegenseitige Konfliktsituationen zu vermeiden (siehe dazu Kapitel 6.11). Joystickbewegungen werden im Normalfall anhand der Veränderungen im Bild bewertet.

6.4.2 Zurück / Weiter

Bei der Einrichtung einer Einstellung werden oft Kommandos verwendet, die eine Fahrt auf die letzte Position zurück oder eine Bewegung in die vorhergehende Richtung fordern. Abbil-

Abbildung 6.7 zeigt den schematischen Ablauf eines verringernden *zurück*-Kommandos anhand des Beispielsatzes *Kamera1 fahre ein wenig zurück*. Ein *weiter*-Kommando würde sich äquivalent invers verhalten.

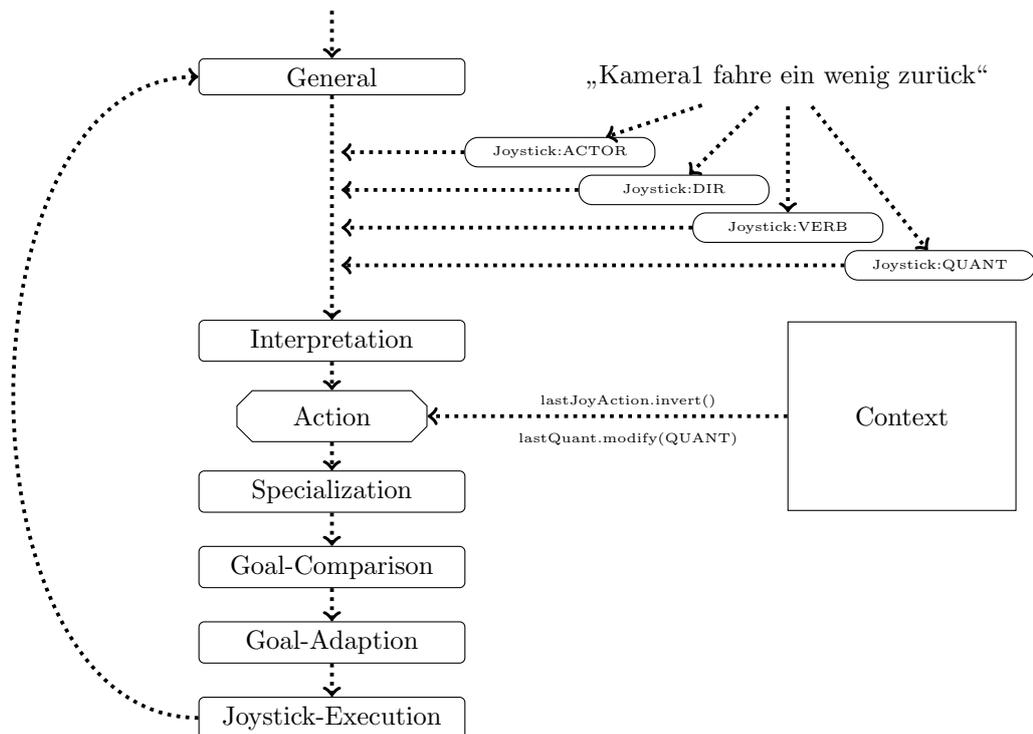


Abbildung 6.7: Schematischer Ablauf eines verringernden *zurück*-Kommandos. Die logischen Fragmente aus der Sprachverarbeitung werden in der *Interpretation*-Schicht in eine *Action* zusammengefasst. Die *Action* generiert aus der Quantitätsinformation des Kommandos zusammen mit Informationen aus dem *Context* eine neue Fahrweisung, die die weiteren Schritte der Reflect-Schicht durchläuft.

Die logische Form des in Abbildung 6.7 dargestellten Kommandos gliedert sich in vier Fragmente: Den Adressaten **ACTOR**, die Richtung **DIR**, das Verb **VERB** und die Quantität **QUANT** der Aussage. **ACTOR** repräsentiert das angesprochene Kamera-System. Entspricht der Akteur nicht dem Aktuellen, wird das Kommando ignoriert. **DIR** beschreibt die Richtung, in der weiter- oder zurückgefahren werden soll. **VERB** erfüllt hier keine Funktion und ist nur der Vollständigkeit halber angegeben. Die Quantität bestimmt, wie weit, in Bezug auf die vorhergehende Aktion, vor- bzw. zurückgefahren werden soll. Ist eine Quantität **WENIG** angegeben, fährt das System jeweils die halbe Strecke, für **VIEL** fährt das System jeweils die doppelte Strecke zurück bzw. weiter. Auf diese Weise kann jeder Punkt angefahren werden. Die Quantität bestimmt jeweils das Intervall

6. DAS AKTIONSMODELL ZUR HANDLUNGSGENERIERUNG

und mit Hilfe der Angaben *weiter* bzw. *zurück* kann die Richtung festgelegt werden.

Anhand der vier übergebenen Fragmente wird in der *Interpretation*-Schicht die letzte verfügbare Aktion aus einer Datenbasis **Context** entnommen und bei *zurück* invertiert bzw. für *weiter* erneut ausgeführt. Zusätzlich wird ein Faktor berechnet, der sich abhängig von der übergebenen Quantität entweder halbiert oder verdoppelt und so die jeweilige Fahrdistanz festlegt.

Bei *zurück* und *weiter* gibt es zwei Möglichkeiten diese Kommandos auf Aktionen abzubilden. Entweder man speichert die letzte Position des Kamerasystems oder man speichert die durch die letzte Aktion durchgeführte Bildänderung, um anhand dieser Daten die letzte Position wiederherzustellen (*zurück*) bzw. die letzte Aktion zu wiederholen (*weiter*). Im vorliegenden System wurde die zweite Möglichkeit präferiert, weil bei diesem Ansatz, auch bei einem geänderten Welt-Zustand, immer noch sendefähige Bilder generiert werden können. Es werden also bei den Kommandos *zurück* bzw. *weiter* die letzten Bildänderungen invertiert bzw. weiter fortgeführt. Unter Bildänderungen verstehen wir hier z.B. die Veränderungen der Position oder der Größe von Objekten. Bei *zurück* nach einer Pan-Bewegung wird das Objekt entsprechend an die letzte Position im Bild zurück verschoben. Bei *weiter* erfolgt die Verschiebung über das zuletzt erreichte Ziel hinaus. Der Vorteil dieser Herangehensweise liegt in der Berücksichtigung der jeweiligen Bildeigenschaften. Dies gilt auch, wenn sich während der Rückfahrt die Welt verändert, also z.B. der Moderator vom Stand- auf das Spielbein wechselt und sich so die Position des Oberkörpers ändert. Außerdem entspricht es der menschlichen Vorgehensweise, wenn *zurück*- bzw. *weiter*-gefahren werden soll.

6.5 Bildausschnittsbestimmende Kommandos

Die *Einstellungsfindung* beginnt mit dem Ausrichten der Kamera auf die Zielobjekte. Hierbei werden bildausschnittsbestimmende Kommandos verwendet. Sie verändern die Achsen Pan, Tilt und Zoom so, dass die gewünschten Objekte im Bild zu sehen sind. Abbildung 6.8 zeigt den schematischen Ablauf anhand des Beispielsatzes: *Kamera1 zeige den Moderator*.

Die Eingabe wird durch die logischen Fragmente **ACTOR**, **VERB** und **ENUMERATION** an das System übergeben. Jedes einzelne Fragment ist dabei durch den Typ **Show** (vgl. Kapitel 5.3.1) markiert. **ACTOR** und **VERB** stehen für den Adressaten sowie für das verwendete Verb. **ENUMERATION** ist eine Aufzählung aller in der natürlichen Sprache referenzierten Objekte.

Nachdem die Standard-Überprüfungen in der *General*-Schicht abgeschlossen und keine fehlenden Informationsquellen festgestellt wurden, können die Fragmente in der *Interpretation*-

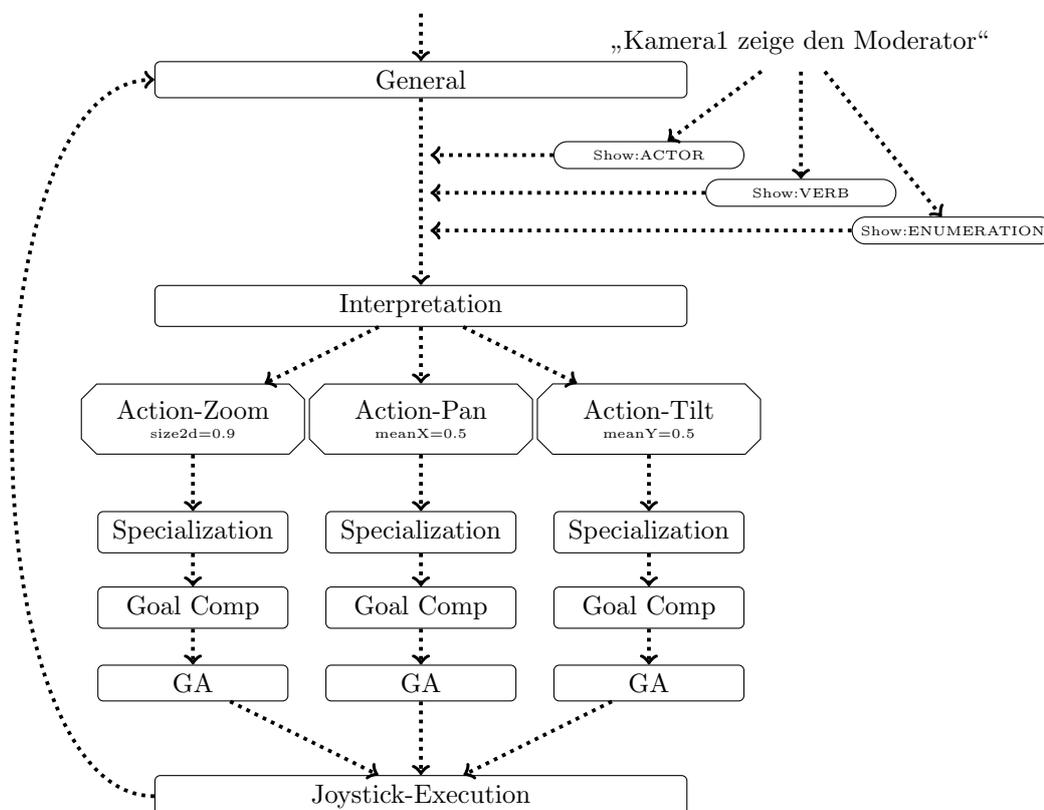


Abbildung 6.8: Schematischer Ablauf eines bildausschnittsbestimmenden Kommandos. Die *Interpretation*-Schicht generiert aus den übergebenen logischen Fragmenten drei *Actions*. Mit Hilfe dieser *Actions* wird sichergestellt, dass die aufgezählten Objekte (hier nur der Moderator) im Bild zu sehen sind. Die Außenkanten der Objekte sollen insgesamt 90% des Bildes einnehmen (*Action-Zoom*). Der Mittelpunkt der Objekte sollte sich sowohl horizontal als auch vertikal in der Mitte des Bildes befinden. Dies wird durch *Action-Pan* und *Action-Tilt* erreicht.

Schicht ins System übernommen und interpretiert werden. Dabei werden drei Aktionen *Action-Zoom*, *Action-Pan* und *Action-Tilt* generiert.

Action-Zoom sorgt dafür, dass die Darstellung aller im Kommando genannten Objekte die Größe von ca. 0.9 erreicht (6.3). Das Bild der Kamera ist vertikal und horizontal in das Intervall $[0; 1]$ eingepasst. Die Geschwindigkeit des Zoom-Joysticks berechnet sich aus der Differenz d der tatsächlichen Größe der Objektgruppe und dem Wert 0.9.

$$v_{Zoom} = \begin{cases} +d & \text{falls } size2d(\text{allObjects}) \leq 0.9 \\ -d & \text{falls } size2d(\text{allObjects}) > 0.9 \end{cases} \quad (6.3)$$

Die Gesamtgröße der Objektgruppe wird mit Hilfe des Prädikats *size2d* ermittelt. Für den

6. DAS AKTIONSMODELL ZUR HANDLUNGSGENERIERUNG

Fall, dass alle Objekte im Bild sichtbar sind, wird die Gesamtgröße anhand der Minima und Maxima der jeweiligen Ränder eines Objekts ermittelt. *size2d* berechnet jeweils getrennt die vertikale und horizontale Ausdehnung der Gruppe und gibt dann das Maximum dieser beiden Werte zurück. PseudoCode 4 zeigt die Berechnung dieses Prädikats. Über alle aufgezählten Objekte, wird der jeweilige Minimal- und Maximalwert für die Position der Außenkanten bestimmt. Daraus wird die maximale Ausdehnung der Enumeration bestimmt.

PseudoCode 4 *size2d*

```
1: MinMaxEvaluator evalh;
2: MinMaxEvaluator evalv;
3: for all selected Objects o do
4:   evalh.update(o.left);
5:   evalh.update(o.right);
6:   evalv.update(o.top);
7:   evalv.update(o.down);
8: end for
9: diffh  $\rightarrow$  evalh.getMax() – evalh.getMin()
10: diffv  $\rightarrow$  evalv.getMax() – evalv.getMin()
11: return max(diffh, diffv);
```

Sind die Objekte nicht sichtbar, muss mit Hilfe eines Kameramodells die jeweilige Abbildungsposition des Objekts auf dem Bild berechnet werden oder mittels logischen Schließens iterativ die Einstellung so lange verändert werden, bis alle Objekte sichtbar sind. Eine reale Kamera kann durch ein projektives Kameramodell abgebildet werden. Nach [HZ04] wird hier jeder Punkt X aus R^3 durch eine Projektionsmatrix P auf den Bildpunkt x abgebildet. $x = PX$ beschreibt diese Abbildung. $P = KR[I - C]$ wird durch das Produkt der Kameramatrix K , der Orientierung der Kamera R sowie der Mischmatrix $[I - C]$ aus der Identitätsmatrix I und dem Vektor der Kameraposition C repräsentiert. In (6.4) bis (6.6) sind die für P notwendigen Matrizen und Vektoren dargestellt.

$$K = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6.4)$$

$$R = R_x R_y R_z \quad (6.5)$$

$$C = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \quad (6.6)$$

K wird als Kameramatrix bezeichnet und beschreibt die Abbildung der Kamera. Neben der Fokusslänge α_x und α_y gehen ein Verzerrungskoeffizient s und der Zentralversatz (Centershift) x_0 und y_0 in diese Matrix ein. R besteht aus drei Rotationsmatrizen R_x , R_y und R_z , die jeweils eine Drehung um die entsprechende Achse beschreiben. C beschreibt den Positionsvektor der Kamera durch die Komponenten x_c , y_c und z_c . Die Projektionsmatrix erlaubt es, auch Objekte abzubilden, die nicht sichtbar sind. So können die oben vorgestellten Prädikate auch für nicht sichtbare Objekte bestimmt werden.

Action-Pan stellt sicher, dass die Kamera auf die Mitte der Gruppe schaut. Ähnlich wie *size2D*, berechnet *meanX* die horizontale Position auf der Basis des Intervalls $[0;1]$. Ziel der Aktion ist es, die Kamera so auszurichten, dass der errechnete Punkt in der Mitte des Bildes ($\text{meanX} = 0.5$) liegt. *Action-Tilt* übernimmt eine vergleichbare Funktion in vertikaler Orientierung.

6.6 Kommandos zur Positionierung von Objekten

Unter Positionieren wird eine horizontale Verschiebung von Objekten im Bild verstanden. Die vertikale Positionierung von Objekten geschieht durch den Begriff *Luft* und wird in Kapitel 6.7 vorgestellt. Die Möglichkeit einzelne Objekte verschieben zu können, erleichtert das Einrichten von Einstellungen. Für Nachrichten- und einfache Magazinsendungen bezieht sich dieses Kommando meist auf die Position des Moderators. Seine funktional wichtigsten Positionen sind neben der Position in der Mitte des Bildes die rechte oder die linke Bildseite. Das Steuerungssystem ist auch in der Lage beliebige Objekte auf beliebige Positionen zu setzen. Dafür muss im Kommando die gewünschte Position angegeben werden. Für dieses Kommando werden dem System vier Fragmente übergeben: **ACTOR**, **VERB**, **PATIENT** und **POSITION**.

ACTOR beschreibt die angesprochene Kamera, das **VERB** das jeweils benutzte Verb. Für das Kommando kommt für **VERB** nur *setze* in Frage. **PATIENT** beschreibt das jeweilige Objekt und **POSITION** die Position, auf die das Objekt verschoben werden soll. Alle Fragmente sind durch den Typ des Kommandos **COA** (ComposeObjectAdj) markiert.

Die vier Fragmente werden in der *Interpretation*-Schicht in einer *Action* vereint. In der *Specialization*-Schicht werden die quantitativen Positionsangaben in die entsprechenden Werte 0.25 für den linken Rand, 0.5 für Mittig und 0.75 für den rechten Rand verwandelt. Die Grundlage dieser Festsetzung ist die Drittel-Regel. Aus der ComposeObjekt-Aktion wird eine Joystick-Aktion der Achse Pan generiert. Diese wird dann in den darunter liegenden *Goal-Comparison*-

Goal-Adaption- und *Joystick-Execution-*Schicht ausgeführt. Auf eine grafische Darstellung wurde verzichtet, weil der Ablauf der Abbildung 6.9 gleicht.

6.7 Kommandos zur Raumänderung von Objekten

Raumändernde Kommandos werden durch die Eigenschaft *Luft* bzw. durch deren Gegenteil *Präsenz* ausgedrückt und können unterschiedliche Aktionen nach sich ziehen. Wird für die Eigenschaften *Luft* keine Seite angegeben, muss die Kamera vom Objekt wegbewegt werden. Der Bezeichner *Präsenz* löst eine Bewegung zum Objekt hin aus. Wird für den Bezeichner *Luft* zusätzlich eine Seite angegeben, soll im Bild auf der jeweiligen Seite des Objekts etwas mehr Platz eingerichtet werden. *Präsenz* schließt eine Verbindung mit einer Seite aus. Abbildung 6.9 zeigt die Verarbeitung von raumändernden Kommandos.

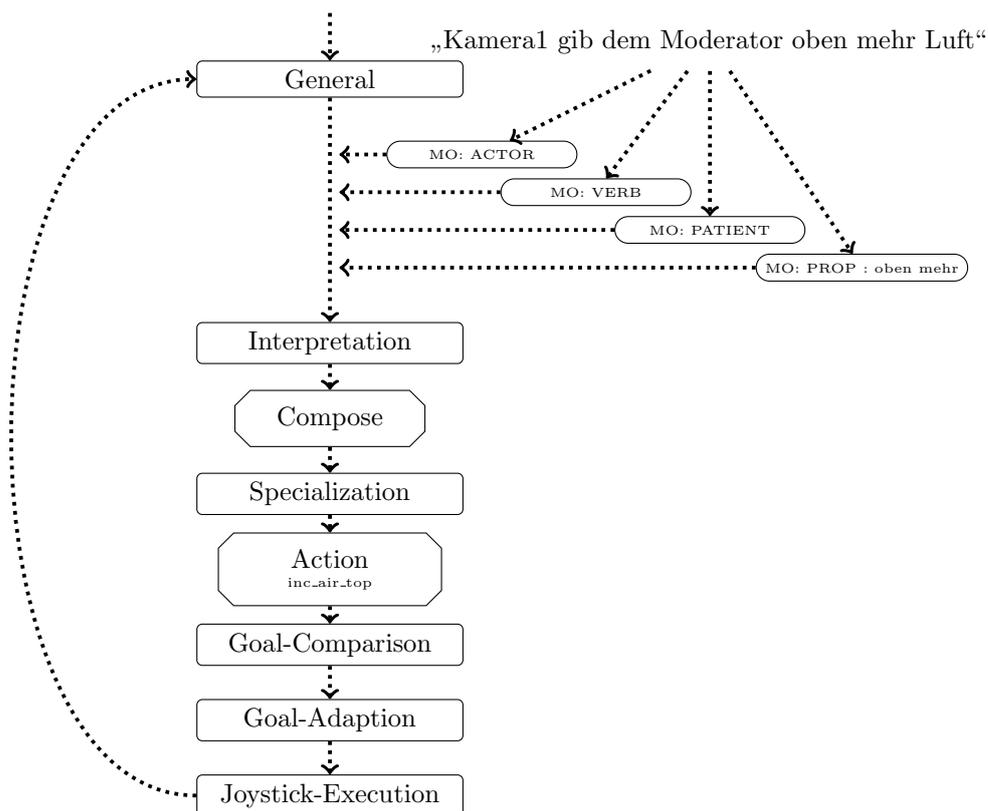


Abbildung 6.9: Schematischer Ablauf eines raumändernden Kommandos. Die Fragmente der Sprachverarbeitung werden in der *Interpretation*-Schicht an die Wissensbasis übergeben. In der *Specialization*-Schicht wird die jeweilige Wortkombination auf eine *Action* abgebildet, die dann weiter ausgeführt wird.

6.7 Kommandos zur Raumänderung von Objekten

Aus der Sprachverarbeitung werden für dieses Kommando vier Fragmente **ACTOR**, **VERB**, **PATIENT** und **PROP** geliefert. Das Fragment **ACTOR** beschreibt die jeweils angesprochene Kamera, **VERB** das verwendete Verb. **PATIENT** beschreibt das Objekt und **PROP**, wie dieser Raum verändert werden soll. **PROP** kann dabei sowohl Quantitätsattribute, wie *mehr* und *weniger*, als auch Positionsattribute (*oben*, *unten*, *rechts*, *links*) enthalten. Zusätzlich enthält **PROP** das jeweils zugewiesene Eigenschaftsnomen *Luft* oder *Präsenz*. Alle Fragmente sind in Abbildung 6.9 mit **MO** (ModifyObject) markiert.

Die Verarbeitung der Aktionen geschieht wie in Kapitel 6.6 bereits beschrieben. In der *Interpretation*-Schicht werden die logischen Fragmente in das System integriert. Es wird vom System eine ModifyObject-Aktion angelegt, die die übergebenen Informationen aufnimmt. In der *Specialization*-Schicht wird dieses Hauptziel (Task) auf (mehrere) Unterziele (Aktionen) aufgliedert, die dann in den weiteren *Goal-Comparison*-, *Goal-Adaption*- und *Joystick-Execution*-Schichten weiterverarbeitet werden.

Die übergebene Kombination aus **VERB** und **PROP** beeinflusst die Achse, die das System verwendet. Enthält **PROP** keine Positionsangabe, kann das System anhand des Verbs und evtl. vorhandener Quantitätsattribute zwischen *näher ran* und *weiter weg* unterscheiden. Für das Verb *geben* wird in Zusammenhang mit dem Wort *Luft* der Abstand zum Rand vergrößert. Für *Präsenz* wird der Abstand verringert. Kommt in der Anweisung ein *weniger* vor, invertiert sich diese Zuordnung. Die Kombination aus *nehmen* und *weniger* kann nicht vorkommen. Tabelle 6.1 zeigt diesen Zusammenhang.

Tabelle 6.1: Zuordnung Verben und Adverbien auf Luft-Eigenschaften

	mehr		weniger	
	Luft	Präsenz	Luft	Präsenz
geben	INC_{AIR}	DEC_{AIR}	DEC_{AIR}	INC_{AIR}
nehmen	DEC_{AIR}	INC_{AIR}	–	–

Enthält **PROP** eine Richtungsangabe, kann nur noch *Luft* angeführt sein. *Präsenz* gilt immer allgemein für ein bestimmtes Objekt und kann nicht in Zusammenhang mit einer Richtung verwendet werden. Für die Richtungen *oben* bzw. *unten* wird entsprechend den optionalen Attributen *mehr* bzw. *weniger* die Achse Tilt verwendet. Für die Richtungen *rechts* bzw. *links* wird die Kamera entsprechend dem Kommando in Abhängigkeit vom Attribut *mehr* bzw. *we-*

niger verschoben. Wird kein Quantitätsattribut angegeben, wird das gleiche Verhalten wie für das Attribut *mehr* ausgeführt.

6.8 Konditional-Kommandos

Konditional-Kommandos geben eine Aktion vor, die bis zum Eintreten einer bestimmten Eigenschaft durchgeführt werden soll. Sie werden durch die Sprachverarbeitung in zwei Fragmente ACTION und CONDITION zerlegt, die jeweils entsprechend viele Unter-Fragmente enthalten. Im Beispiel *Kamera1 fahre nach links bis das Objekt nicht mehr zu sehen ist* (vgl. Abbildung 6.10) enthält der ACTION-Teil des Satzes die Unter-Fragmente der Joystick-Bewegung (vgl. Kapitel 6.4). Der CONDITION-Teil des Satzes enthält die Unter-Fragmente PATIENT, PROP und NEG. PATIENT beschreibt das Objekt, für das die Eigenschaft gilt, PROP beschreibt die Eigenschaft selbst und NEG eine evtl. vorhandene Negation. Abbildung 6.10 zeigt schematisch den Ablauf von Konditional-Kommandos.

Diese Art von Kommandos wird aufgrund ihrer Komplexität für jeden Fall unterschiedlich behandelt. Die bis zum Zeitpunkt des Schreibens dieser Arbeit unterstützten Kommandos beziehen sich auf Sichtbarkeit, Nicht-Sichtbarkeit und Kontakt zwischen einzelnen Objekten. In Abbildung 6.10 werden die übergebenen Fragmente als *Action* in das System integriert. In den nachstehenden Schichten kommen für den Aktionsteil die gleichen Überprüfungen zum Tragen wie für „einfache“ Kommandos. Einzig in der *Goal-Comparison*-Schicht muss die angefragte Eigenschaft bekannt sein.

Konditional-Kommandos, die sich auf die Sichtbarkeit beziehen, prüfen, ob sich die angegebenen Objekte innerhalb des sichtbaren Bereichs befinden. Dabei wird getestet, ob sich die Position des linken und des rechten Rands des Objekts innerhalb des Intervalls $[0; 1]$ befinden. Ist dies der Fall, kann die Aktion abgebrochen werden. Im anderen Fall muss die Aktion in einem weiteren Zyklus ausgeführt werden. Die formale Beschreibung zeigt (6.7).

Für Kommandos, die sich auf Nicht-Sichtbarkeit beziehen, wird geprüft, ob sich das Objekt außerhalb des sichtbaren Bereichs befindet. Dafür muss sich der linke Rand des Objekts „rechts“ vom rechten Bildrand oder der rechte Rand des Objekts „links“ vom linken Bildrand befinden. (6.8) zeigt dies in formaler Darstellung.

$$P_{\text{sichtbar}} : (x < (1 - 0.5 \cdot w)) \wedge (x > 0.5 \cdot w) \quad (6.7)$$

$$P_{\text{unsichtbar}} : ((x + 0.5 \cdot w) < 0) \vee ((x - 0.5 \cdot w) > 1) \quad (6.8)$$

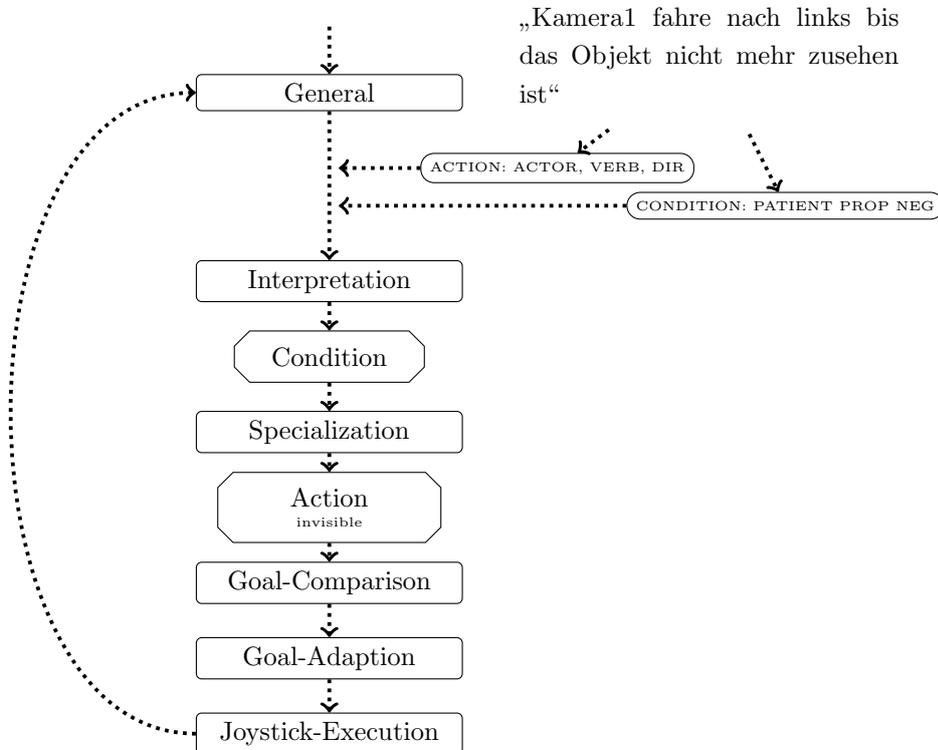


Abbildung 6.10: Schematischer Ablauf einer Konditional-Anweisung. In der *Interpretation*-Schicht werden die Fragmente des Kommandos in die Wissensbasis aufgenommen und ein *Condition*-Task erzeugt. In der *Specialization*-Schicht wird aus diesem eine *Action* für die jeweilige Kondition des Kommandos generiert. Diese *Action* durchläuft dann die weiteren Schichten der Verarbeitung.

Bei Kommandos, die sich auf den Kontakt zwischen zwei Objekten beziehen, prüft das System für die jeweils übergebene Achse den jeweils minimalen Abstand zwischen allen relevanten Seiten. Für die Achse Tilt und Z werden die Abstände der Ober- und Unterseite berechnet, für alle anderen Achsen werden die Abstände der rechten und linken Seite zwischen den Objekten berechnet. Ist der Abstand zwischen den Seiten für zwei Objekte Null¹, gilt die Aktion als ausgeführt. Steht beim Initialisieren der Aktion bereits einer der Abstände auf Null, wird solange weiter gefahren, bis ein anderes Abstandspaar Kontakt zueinander bekommt.

¹oder nahe Null

6.9 Speichern-Kommando

Das Speichern bietet die Möglichkeit den aktuellen Zustand festzuhalten, um zu einem späteren Zeitpunkt an gleicher Stelle anknüpfen zu können. Außerdem werden mit Hilfe des Kommandos die Basis-Einstellungen für die *Sendekontrolle* festgelegt. Zu diesem Zweck werden neben den Positionsdaten auch die aktuellen Bildeigenschaften gesichert. So kann beim Wiederanfahren auch bei einem veränderten Set ein gleicher Bildeindruck wiederhergestellt werden. Das Speichern-Kommando kann sowohl mit Angabe eines Speicherziels als auch ohne diese Angabe an das System übergeben werden. Ist ein Speicherziel genannt, werden die Informationen in diesem Speicherziel gespeichert. Fehlt die Angabe, generiert das System automatisch einen Namen, unter dem die Einstellung gespeichert wird.

6.10 Anfahren einer bestimmten Positionen und Einstellung

Das vorgestellte System erlaubt zwei unterschiedliche Verwendungen dieser Kommandos. Zum einen können vordefinierte Standard-Einstellungen und zum anderen können zuvor gespeicherte Positionen angefahren werden.

Unter vordefinierten Einstellungen werden allgemeingültige Definitionen von Einstellungen wie *Totale*, *Halbnahe* oder *Nahe* verstanden. Sie sind der Startpunkt bei der *Einstellungsfindung* und sind sendungsübergreifend abgestimmte Standards, die allgemein bekannt sind (vgl. Kapitel 3.1.2). Gesicherte Positionen sind speziell für eine Sendung und einen bestimmten Moderator angepasste Einstellungen. Auf der Basis der gespeicherten Positionen werden später die jeweiligen Sendungen durchgeführt. Beide Funktionen werden über das gleiche Kommando angefragt. Anhand des übergebenen Namens unterscheidet das System, ob eine allgemeine oder eine spezialisierte Einstellung gemeint ist.

Durch die Sprachverarbeitung werden drei logische Fragmente an das System übergeben: **Goto:ACTOR**, **Goto:VERB** und **Goto:PATIENT**. Das Fragment **ACTOR** legt das angesprochene Kamerasystem fest. **VERB** gibt das Verb wieder. **PATIENT** beschreibt nominell, was geladen werden soll. System-Intern werden alle Kommandos als Goto-Kommandos bezeichnet. Deshalb sind alle drei Fragmente mit dem Tag **Goto** markiert. In der *Interpretation*-Schicht wird die Goto-Anweisung in das System aufgenommen. Hier wird überprüft, ob es sich um eine allgemeine Einstellung (Standard-Shot) oder eine gespeicherte Position (Saved-Shot) handelt. Liegt eine

6.10 Anfahren einer bestimmten Positionen und Einstellung

allgemeine Einstellung vor, fährt das System auf diese Einstellung. Tabelle 6.2 zeigt die jeweiligen Standardwerte, die in der *Specialization*-Schicht jeweils verwendet werden.

Tabelle 6.2: Standardwerte für die *Nahe*, *Halbnahe* und *Amerikanische-Einstellung*

	Z	Zoom	Pan	Tilt
zentrale <i>Nahe</i>	eyeHeight	height2d=0.42	pos_hor=0.5	toppos=0.05
rechte <i>Nahe</i>	eyeHeight	height2d=0.42	pos_hor=0.75	toppos=0.05
linke <i>Nahe</i>	eyeHeight	height2d=0.42	pos_hor=0.25	toppos=0.05
zentrale <i>Halbnahe</i>	eyeHeight	height2d=0.3	pos_hor=0.5	toppos=0.05
rechte <i>Halbnahe</i>	eyeHeight	height2d=0.3	pos_hor=0.75	toppos=0.05
linke <i>Halbnahe</i>	eyeHeight	height2d=0.3	pos_hor=0.25	toppos=0.05
zentrale <i>Amerikanische</i>	eyeHeight	height2d=0.16	pos_hor=0.5	toppos=0.08
rechte <i>Amerikanische</i>	eyeHeight	height2d=0.16	pos_hor=0.75	toppos=0.08
linke <i>Amerikanische</i>	eyeHeight	height2d=0.16	pos_hor=0.25	toppos=0.08
<i>Totale</i>	–	size2D=0.9	mean2D=0.5	–

Sie wurden in Kapitel 3.3.2 bereits eingeführt. Für *Nahe*- und *Halbnahe-Einstellungen* wird die Kamera automatisch auf Augenhöhe gestellt. Die Größe des Moderators sollte für die *Nahe* ca. 42% der Bildschirmhöhe und für eine *Halbnahe* ca. 30% sein. Die einzelnen Standard-Einstellungen sind über die Achsen Z, Pan, Tilt und Zoom definiert. Die Weltachsen X und Y werden nicht berücksichtigt. So kann auf jeder Position eine bestimmte Standard-Einstellung angefahren werden. Prinzipiell erlaubt das System aber die Definition von beliebigen Vorgaben auf beliebigen Achsen. Abbildung 6.11 zeigt den Ablauf für das Kommando *Kamera1 fahre auf eine Halbnahe*.

Im vorgestellten Beispiel handelt es sich um eine allgemeine Einstellung, die in der *Specialization*-Schicht weiter verfeinert wird. Für die hier übergebene zentrale *Halbnahe* sind folgende Eigenschaften voreingestellt:

- Die Kamera befindet sich auf Augenhöhe des Moderators.
- Der Zoom ist so eingestellt, dass die Höhe des Moderatorkopfs ca. 30% der vertikalen Bildfläche bedeckt.
- Die Kamera ist so orientiert, dass sich der Moderator in der Mitte des Bildes befindet:
 $p = 0.5$
- Der Moderator hat ca. 5% *Luft* über dem Kopf.

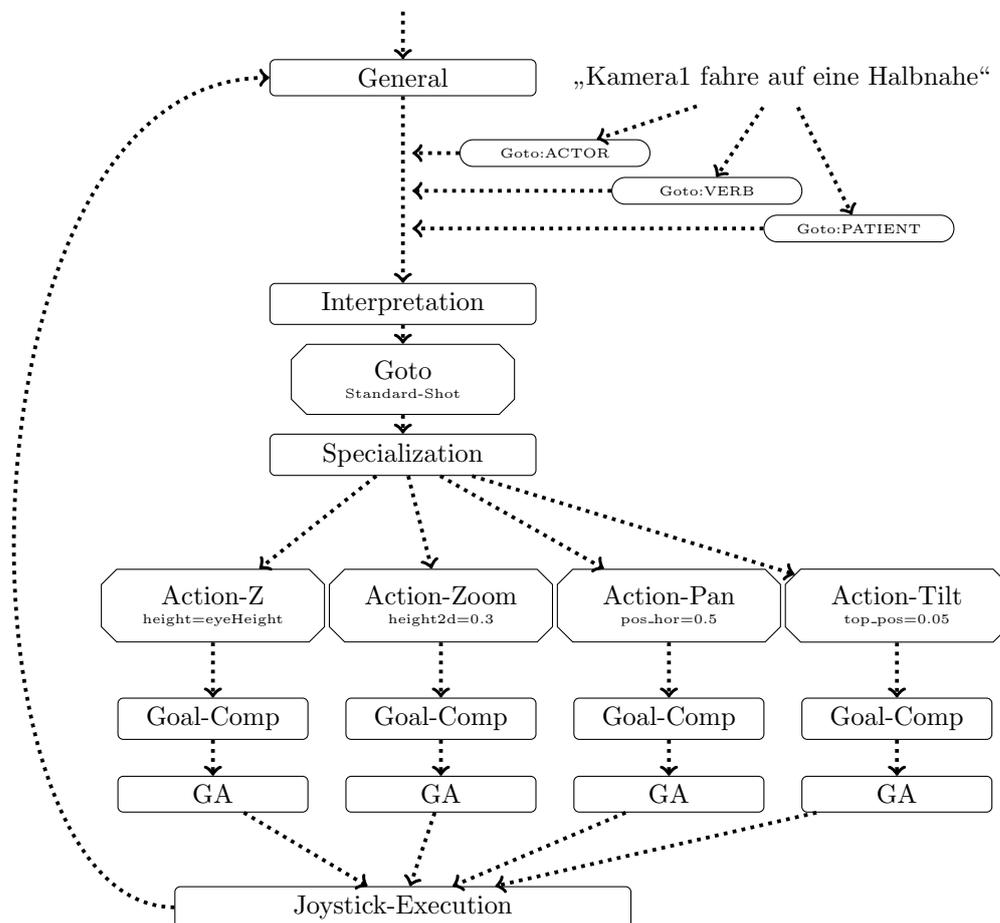


Abbildung 6.11: Schematischer Ablauf für eine Fahrt auf eine Einstellung. Die übergebenen Fragmente werden in der *Interpretation*-Schicht in das System übernommen und es wird ein *Goto*-Task generiert. Aus diesem werden in der *Specialization*-Schicht die, für die jeweilige Einstellung nötigen, Vorgaben in Form von *Actions* generiert.

Die einzelnen Aktionen durchlaufen wieder die *Goal-Comparison*- und *Goal-Adaption*-Schichten, bevor sie an die *Joystick-Execution*-Schicht übergeben werden.

Im Fall, dass keine Standardposition angefahren wird, lädt das System neben den Positionsinformationen auch die gespeicherten Bildinformationen. Sie enthalten die Position und Größe der jeweils auf dem Bild dargestellten Elemente. So kann die zuvor gespeicherte Position auf die aktuelle Situation mit dem aktuellen Moderator angepasst werden (vgl. Kapitel 4.3.2). Gleichzeitig mit dem Anfahren der Position im Studio werden Joystick-Aktionen generiert, die die Größe und Positionen der darzustellenden Objekte im Bild, die Luft über dem Kopf des Mo-

derators sowie die Differenz zwischen Moderator und Stack¹ anpassen. Dabei wird die Größe der darzustellenden Objekte durch die Achse Zoom an die allgemeinen Vorgaben angepasst. Die Position der darzustellenden Objekte wird durch die Achse Pan, die Luft über dem Kopf des Moderators durch die Achse Tilt und der vertikale Abstand zwischen der Oberkante des Moderators und des Stacks durch die Achse Z ausgeführt.

Die Position der Kamera bestimmt die Perspektive auf die Szene. Sie kann durch die reine 2D-Information nicht gespeichert werden, deshalb werden die Achsen X und Y beibehalten. Für die Anpassung müssen die typischen Elemente wie Moderator und Stack in unterschiedlicher Tiefe im Bild liegen. Ist dies der Fall, kann für einen größeren bzw. kleineren Moderator das Bild durch Höhenänderung sehr nah an die Vorgabe angepasst werden. Dazu fährt die Kamera so lange hoch, bis der Abstand der Oberkanten so groß ist, wie beim Speichern des Bildes. Mit Hilfe des Tilts kann dann die vertikale Position der Objekte so angepasst werden, dass die Einstellung identisch zur Vorlage erscheint. Der einzige erkennbare Unterschied ist, dass der Moderator je nach Größe weniger bzw. mehr *Präsenz* im Bild erhält.

6.11 Zeitwährende Kommandos

Unter zeitwährenden Kommandos werden in der vorliegenden Arbeit alle Kommandos verstanden, für die kein Ausführungsende festgelegt ist. Die Kommandos dürfen mit Erreichen des Ziels nicht aus dem System entfernt werden, sondern bestehen so lange weiter, bis sie explizit durch ein *Stopp*-Kommando beendet werden. Mit Hilfe der zeitwährenden Kommandos können bestimmte Bildeigenschaften gefordert und über einen längeren Zeitraum beibehalten werden. So kann z.B. während einer Sendung eine bestimmte Einstellung beibehalten oder in der Phase der *Einstellungsfindung* eine festgelegte Kadrierung konserviert werden, um unterschiedliche Perspektiven erproben zu können. Zeitwährende Kommandos beziehen sich auf bestimmte Standard-Einstellungen oder aber auf bestimmte Kadrierungen im Bild. Sie werden durch das Verb *halte* markiert. Dies veranlasst das System, das Entfernen der jeweiligen Aktionen zu sperren, sodass diese Aktionen im System erhalten bleiben. Abbildung 6.12 zeigt beispielhaft den Ablauf für das Kommando *Kamera1 halte den Moderator mittig*.

Die Eingabe erfolgt ebenso, wie beim Grundkommando aus Kapitel 6.6, durch die vier logischen Fragmente **ACTOR**, **PATIENT**, **VERB** und **PROP**. Das Verb ist hier allerdings nicht *setzte*

¹Stack bezeichnet die Fläche seitlich oder hinter dem Moderator, auf der die zur jeweiligen Meldung passenden Bilder angezeigt werden.

6. DAS AKTIONSMODELL ZUR HANDLUNGSGENERIERUNG

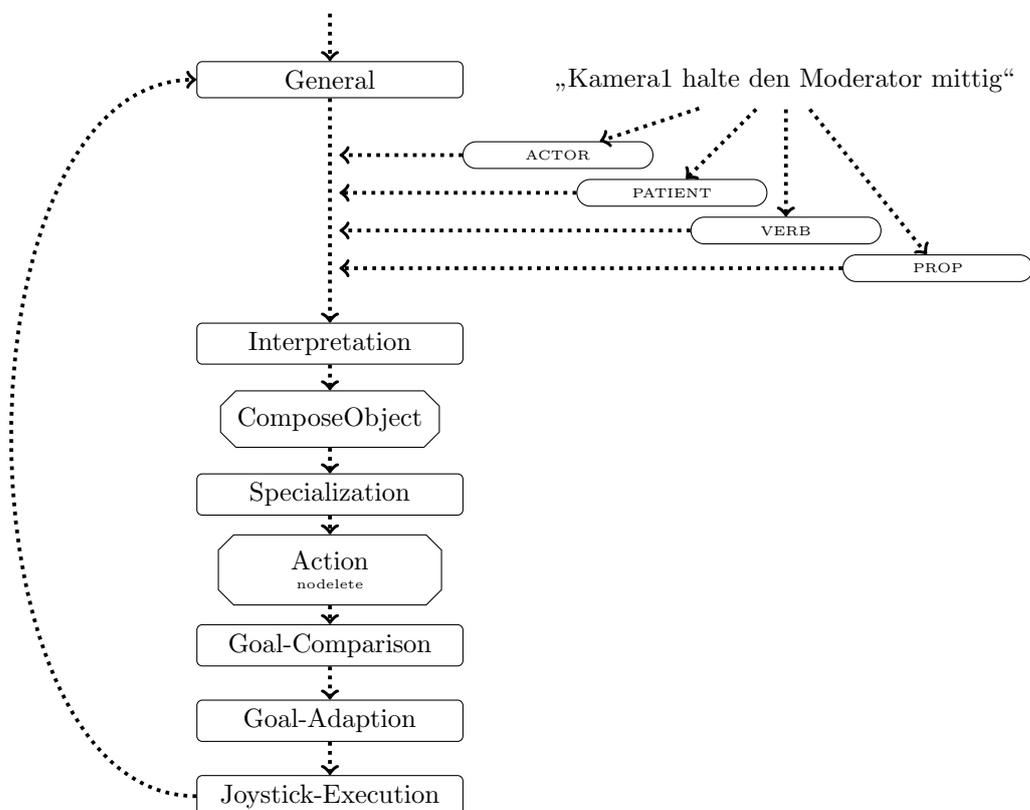


Abbildung 6.12: Schematischer Ablauf eines zeitwährenden Kommandos. In der *Interpretation*-Schicht werden die übergebenen Fragmente in das System integriert. In der *Specialization*-Schicht wird für die geforderte *Action* ein Flag *nodelete* gesetzt, das das Entfernen der *Action* aus dem System verhindert.

sondern *halte*. Während der Bearbeitung der *Action* wird diese als *nodelete* markiert. Die Markierung verhindert, dass die Aktion in der *Goal-Comparison*-Schicht bei Erreichen des Ziels aus dem System entfernt wird.

Wird während eines zeitwährenden Kommandos eine andere Joystick-Aktion gefordert, wird diese gerätebasiert ausgeführt (vgl. Kapitel 6.4.1). Das heißt, abhängig von der Achse wird der Zielpunkt der Aktion in Weltkoordinaten definiert. Nachdem sich sowohl zeitwährende Kommandos als auch Joystick-Kommandos auf Bildeigenschaften beziehen, würden sie bei gemeinsamer Verwendung in Konflikt geraten. Deshalb verwenden Joystick-Kommandos, die während zeitwährenden Kommandos ausgeführt werden, als Bezugssystem den gerätebasierten Ansatz.

Kapitel 7

Implementierung und Evaluation des Prototyps HULICO

In den vorstehenden Kapiteln 3 und 4 wurden Aspekte und Anforderungen beschrieben, die aufgrund von theoretischen und praktischen Erwägungen essenziell für eine intelligente Steuerung eines automatischen Kamerasystems sind. Kapitel 5 stellt eine Kategorial-Grammatik vor, die die im Studio verwendete Sprache in logische Elemente transformieren kann. In Kapitel 6 werden diese Fragmente in das Verhalten des Roboters überführt. Im folgenden Kapitel wird die Implementierung des Prototyps sowie dessen Evaluation vorgestellt. In Kapitel 7.1 wird der Prototyp selbst vorgestellt, der zur Evaluation des Systems entwickelt wurde. Danach werden in Kapitel 7.2 einige Eigenschaften des Prototyps beschrieben und mit den Anforderungen verglichen. In der Evaluation in Kapitel 7.3 wird überprüft und bewertet, inwieweit der in den Kapiteln 4, 5 und 6 beschriebene Ansatz den praktischen Anforderungen entspricht. Außerdem wird durch eine Umfrage festgestellt, ob ein intelligentes, verbal-steuerbares Kamerasystem in einer realen Studioumgebung einen Mehrwert generiert sowie ob der in der vorliegenden Arbeit vorgestellte Satz von Kommandos zur verbalen Führung eines automatischen Kamerasystems ausreicht.

7.1 Prototyp: HULICO

Die in den vorstehenden Kapiteln vorgestellten und beschriebenen Ideen wurden in einem Prototyp HULICO (**H**uman-**L**ike-**C**amera-**O**perator) realisiert. Der Prototyp ist die Grundlage der späteren Evaluation, um die Leistungsfähigkeit und den Nutzen des Systems zu demonstrieren. HULICO besitzt eine natürlichsprachliche Nutzerschnittstelle und kann so wie ein menschlicher

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

Kameramann angewiesen werden. Neben Kommandos zum Anfahren bestimmter Positionen, erkennt HULICO die im Studio etablierten Standard-Einstellungen und kann auf geänderte Bedingungen in der jeweiligen Studioumgebung adäquat reagieren. Durch die Verwendung einer Regelmaschine lässt sich das Wissen und Verhalten von HULICO leicht und schnell für die jeweilig anweisenden Regie-Vertreter anpassen. HULICO setzt dabei auf das Kamerasystem-Interface der RoboKam[©] auf. Die Software für RoboKam[©] wird von der Firma Robotics Technology Leaders GmbH¹ entwickelt. Die RoboKam[©] selbst wird durch die Firma Camerobot GmbH² vertrieben.

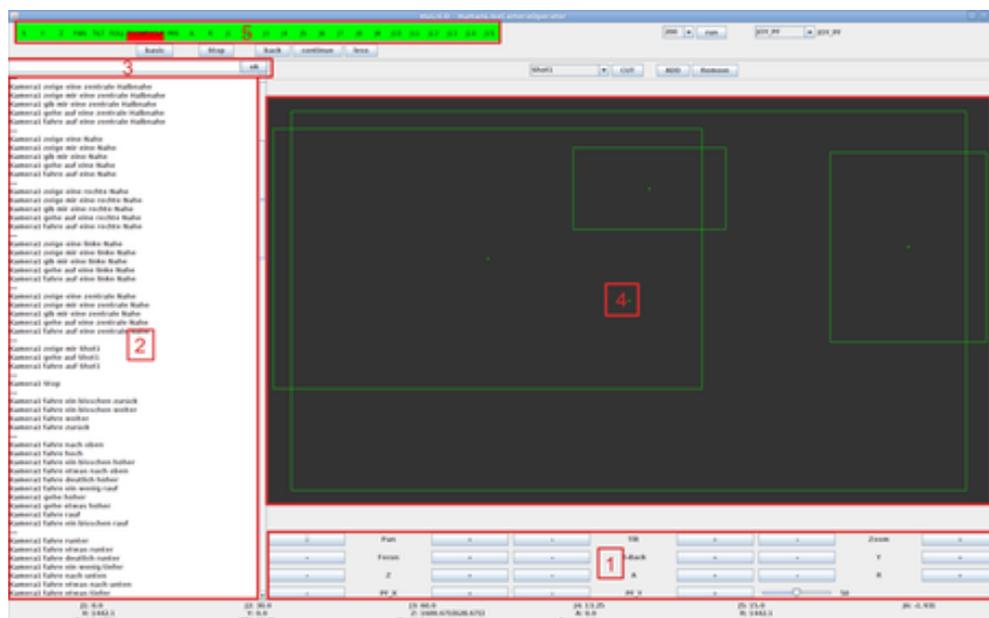


Abbildung 7.1: Grafische Oberfläche des Prototyps HULICO.

Die Oberfläche von HULICO bietet sowohl die Möglichkeit das Kamerasystem (CamSystem) konventionell über einen Software-Joystick (siehe Abbildung 7.1 1) zu steuern, als auch verbal durch Kommandos. Die Kommandos werden in textueller Form an das System übergeben. Auf eine Spracherkennung, die einen gesprochenen Satz in einen Text wandelt, wurde verzichtet, um so fehlerhafte Interpretationen der Kommandos, auf Grund von undeutlicher Aussprache, zu vermeiden. Im Fokus dieses Prototyps stehen die Kommandos selbst. Eine Spracherkennung würde den Aufbau komplizierter machen und birgt ein Fehlerpotential. Einige Standard-Kommandos werden in einer vorbereiteten Liste angeboten (siehe Abbildung 7.1 2). Über ein

¹www.rtleaders.com

²www.camerobot.com

Textfeld (siehe Abbildung 7.1 3) ist es möglich auch beliebige Kommandos an das System zu richten. Außerdem bietet HULICO eine grafische Darstellung der im System vorhandenen „Weltsicht“ (siehe Abbildung 7.1 4). Sie zeigt schematisch alle wesentlichen Elemente des Studiosets. Die Elemente werden durch die Außenmaße ihrer Boundingbox repräsentiert und auf eine schwarze Fläche in der Oberfläche gezeichnet. Die schwarze Fläche entspricht dem aktuellen Bild der jeweiligen Kamera. Das Erreichen der Achsgrenzen (Limits) des verwendeten automatischen Kamerasystems wird ebenfalls in HULICO angezeigt. (siehe Abbildung 7.1 5).

7.1.1 RoboKam[©]

Die RoboKam[©] ist ein modifizierter, kameratragender Industrieroboter vom Typ RX160L der Firma Stäubli (vgl. Abbildung 2.1 (a)). Neben dem Roboter kann die RoboKam[©] auch eine Linse und optionale Zusatzachsen ansteuern. Das System wird bereits in einigen Fernsehstudios eingesetzt und ist dort in den Nachrichtenbetrieb integriert. Zu diesen Studios zählen namhafte deutsche Sender, wie das ZDF, SWR und NDR sowie einige ausländische Sendeanstalten. Die Steuerung des Kamerasystems erfolgt konventionell softwareseitig durch die zur RoboKam[©] gehörende grafische Benutzeroberfläche (GUI) oder hardwareseitig durch Joystick-Konsolen unterschiedlicher Hersteller. Das RoboKam[©]-System besteht aus mehreren Prozessen, die, verteilt auf unterschiedlichen Computern, miteinander interagieren. Die wichtigsten Prozesse sind der MainController und der VR-Prozess sowie das Kamerasystem.

- Der *MainController* ist das Herz von RoboKam[©]. An diesem Prozess registrieren sich alle startenden Prozesse. Jeder Prozess kann so alle nötigen Informationen über das System erfahren und seine entsprechenden Interfaces finden. Gleichzeitig stellt der MainController eine Datenbank für alle Prozesse bereit, in der z.B. Positionsinformationen abgelegt werden können.
- Der *VR-Prozess* dient der Übergabe der Kamera-Pose an das Render-System. Nicht alle RoboKams[©] werden in einem VR-Set eingesetzt. Dieser Prozess ist deshalb für die RoboKam[©] optional. HULICO verwendet ein kommerzielles Render-System der Firma VizRT¹ zur Simulation der Kamera.
- Das *Kamerasystem* (CamSystem) berechnet die Trajektorien für die Kameras. Es nimmt Eingaben der GUI und/oder der Hardware-Joystick-Konsolen entgegen und verrechnet diese mit der Trajektorie. Zudem stellt es ein Interface zu einem *VR-Prozess* zur Verfügung.

¹<http://www.vizrt.com/>

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

Über dieses Interface können die aktuelle Kamera-Pose sowie die Linsenstellung an ein Render-System übergeben werden, sodass dort ein entsprechend passendes virtuelles Bild berechnet und später in das Fernsehbild integriert werden kann.

7.1.2 Schematischer Aufbau

Schematisch lässt sich HULICO auf drei interagierende Komponenten aufteilen: HULICO selbst als Steuerungskomponente, das *Kamerasystem* für die Berechnung der Trajektorien und eine Komponente *World*, die für die Umgebung steht, in der HULICO verwendet wird. Im Rahmen des Testaufbaus besteht die Welt (World) aus einer Simulation des Kamerabildes sowie einem Tracking-System, das Informationen über das Videobild und die reale Welt zur Verfügung stellt. Um HULICO in einem realen Szenario einsetzen zu können, müsste die Simulation getrennt und dafür eine reale RoboKam[©] verwendet werden. Die erforderlichen Informationen müsste ein Tracking-System liefern, das sowohl die Weltposition einzelner Objekte (6D) als auch die Bildinformationen (2D) liefern kann. Ein solches System stand nicht zur Verfügung. Deshalb wurde ein Echtzeitrender-System der Firma VizRT¹ zur Simulation der Bildinformationen eingesetzt. Das Render-System wird sowohl für virtuelle Studios als auch für Virtual-Reality-Effekte verwendet und generiert deshalb eine hochgradig realitätstreue Simulationsumgebung. Das Render-System nimmt die Kamera- und Linsendaten vom Kamerasystem entgegen und berechnet damit das aktuelle Kamerabild. Die nötigen Informationen, wie die Positionsdaten der einzelnen Objekte im Studio, werden durch, im Rahmen der Arbeit entwickelte, Plugins im Render-System generiert und an HULICO weitergereicht. HULICO übernimmt diese Informationen und verwendet sie für die Wissensbasis. Abbildung 7.2 zeigt den schematischen Aufbau von HULICO im Zusammenhang mit der RoboKam[©].

HULICO nutzt zwei Interfaces **CS** und **CBTracking** des RoboKam[©]-Systems. Das Interface **CS** erlaubt HULICO die Position und Orientierung der Kamera sowie die Position der Linse über die Funktion `setJoystickAxisSpeed(vals:Map)` zu steuern. `vals` enthält dabei die Geschwindigkeiten der einzelnen Achsen. Außerdem werden über **CS** kontinuierlich Statusmeldungen des Kamerasystems an HULICO übergeben. Der Status enthält die Positionsdaten der Kamera und der angeschlossenen Geräte, Warnungen beim Erreichen von Limits und Fehlermeldungen.

Über **CBTracking** erhält HULICO Informationen über die einzelnen vorhandenen Objekte in der Welt. Die Daten werden kontinuierlich gesendet und enthalten neben der 3D-Positionsinformation den Namen des jeweiligen Objekts, sowie die Position und Größe im Video-Bild.

¹<http://www.vizrt.com/>

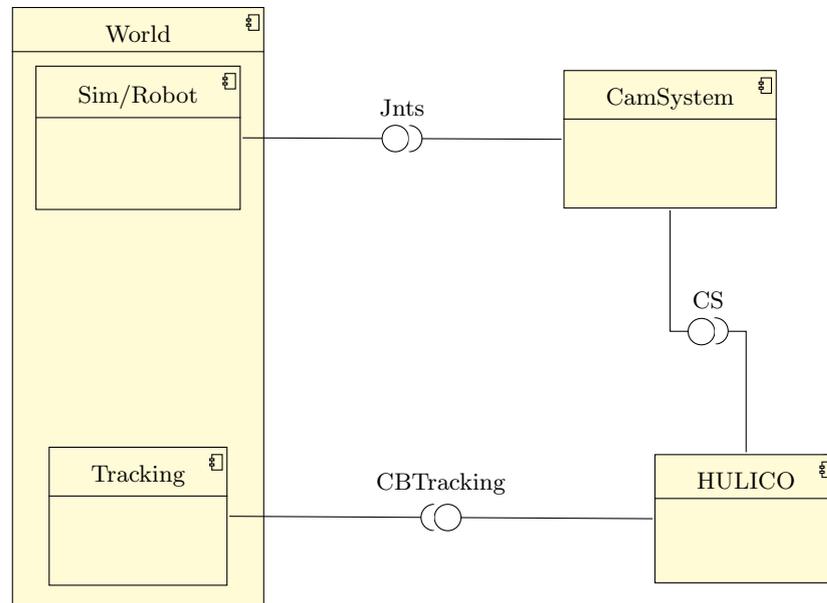


Abbildung 7.2: Komponentendiagramm von HULICO. HULICO nutzt zwei Interfaces zur Interaktion mit der Welt. Das CS-Interface dient der Steuerung des Kamerasystems sowie zum Empfang von kamerasystemspezifischen Daten. Ferner nutzt HULICO das CBTracking-Interface zum Empfangen von Welt-Informationen.

Das Kamerasystem übernimmt die Befehle von HULICO und generiert daraus Stützstellenlisten für das RoboKam[©]-System. Über das Interface **Jnts** werden diese Stützstellen-Informationen an die jeweilig ausführenden Geräte weitergereicht. So entsteht ein geschlossenes Testsystem, das auf zwei Computern (Render-System und HULICO-PC mit RoboKam[©]-Software) betrieben wird.

7.1.3 Realisierung

Das in Kapitel 6 beschriebene Regelsystem wurde für den Prototypen HULICO in JAVA (Version 1.6) mit Hilfe der Rules-Engine Drools (Version 5.5.0) [Bro09] implementiert. Die Hierarchie der Schichten konnte mit Hilfe der Agenda von Drools integriert werden. Die Regeln der *General*-Schicht bekommen eine Gewichtung (Saliency) von 100, Regeln der *Joystick-Execution*-Schicht erhalten eine Gewichtung von 30. Alle dazwischen liegenden Schichten bekommen entsprechen ihrer Reihenfolge Werte in 10er Schritten zugewiesen. Regeln innerhalb einer Schicht bekommen eine entsprechende Priorität innerhalb des für die Schicht gültigen Intervalls. Die Priorität der *Interpretation*-Schicht ist im Prototyp auf das Intervall 75-80 festgelegt. Die Be-

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

handlung des Stopp-Kommandos hat innerhalb dieses Intervalls eine Saliency zwischen 78 und 79 während die anderen Kommandos mit einer Saliency von 75 versehen werden. Das Stopp-Kommandos wird so bei jedem Durchlauf zuerst ausgeführt.

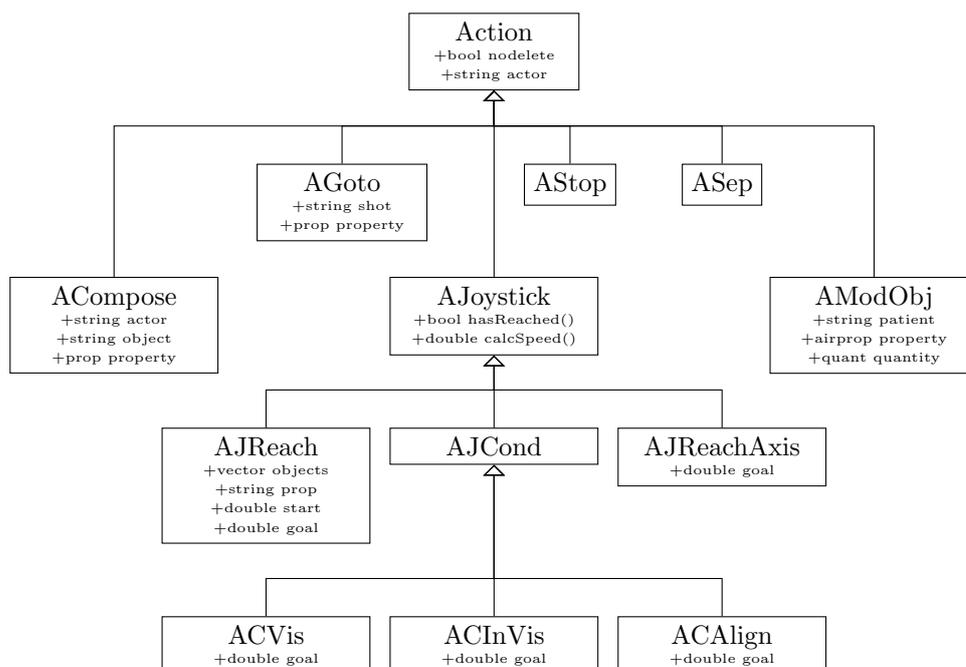


Abbildung 7.3: Klassenhierarchie der Aktions-Klassen. Mit Hilfe dieser Klassen wurde das Verhalten im Rahmen des Prototyps HULICO entwickelt und evaluiert.

Die einzelnen Aktionen sind als Klassen implementiert. Abbildung 7.3 zeigt deren Klassenhierarchie. Die Basisklasse aller Aktionen ist die Klasse **Action**. Sie enthält ein Attribut (`nodelete`), das festlegt, ob die Aktion nach Erreichen des Ziels aus dem System entfernt werden soll oder nicht. Zusätzlich enthält die Aktion den Namen des mit dem Kommando angesprochenen Akteurs.

Die Klasse **ASep** wird als Separator auf den Kontext (siehe Kapitel 6.2.4 und 6.4.2) verwendet. Die Klasse trennt unterschiedliche Aktionen voneinander. So können Tasks, die aus mehreren Aktionen bestehen, rückgängig gemacht (*zurück*) oder erneut ausgeführt werden (*weiter*).

Die Klasse **AStop** repräsentiert das Stopp-Kommando in der Wissensbasis. Das System erkennt anhand dieser Aktion, dass alle anderen laufenden Aktionen entfernt werden müssen.

Die Klasse **AGoto** steht für ein Kommando zur Anfahrt einer Position bzw. Einstellung (Goto-Kommando). Es hält den Bezeichner für das Ziel und optional ein Attribut `property`, das die jeweilige Seite beschreibt. `property` wird nur bei angeforderten Standard-Einstellungen ver-

wendet.

Die Klasse `AModObj` wird für die Modellierung der Kommandos zur Raumänderung von Objekten verwendet. Diese Klasse enthält das Attribut `patient`, für das die Aktion gilt. Ein weiteres Attribut `property` beschreibt, ob mehr oder weniger Raum um dieses Objekt geschaffen werden soll bzw. ob zusätzlich eine bestimmte Richtung angegeben wurde. Abhängig von dieser Information werden unterschiedliche Achsen für die Ausführung der Aktion angesprochen. Das Attribut `quantity` steht für die Quantitätsangabe des Kommandos.

Die Klasse `AJoystick` steht, zusammen mit allen ihren Kind-Klassen, für Bewegungsaktionen des Kamerasystems. Diese Klassen implementieren jeweils zwei Funktionen `hasReached()` und `calcSpeed()`, mit deren Hilfe die neue Geschwindigkeit berechnet wird und festgestellt werden kann, ob das jeweilige Ziel bereits erreicht wurde.

Die Klasse `AJReach` (`ActionJoystickReach`) steht für Aktionen, die einen bestimmten Zustand im Bild erreichen sollen. Dazu wird eine Achse und ein Zielzustand beschrieben, anhand dessen bestimmt werden kann, ob das Ziel bereits erreicht wurde. Die Klasse `AJReachAxis` funktioniert nach dem gleichen Prinzip, allerdings werden hier keine Eigenschaften aus dem Bild als Ziel formuliert, sondern Werte der jeweiligen Achse. Die Klassen um `AJCond` (`ActionJoystickCondition`) führen ebenfalls Bewegungen aus, bis bestimmte Zustände erreicht sind. Allerdings sind die formulierten Zustände Sonderfälle, die eigene Berechnungsmethoden benötigen. Die Klassen `ACVis` (`ActionConditionVisible`) und `ACInVis` (`ActionConditionInvisible`) testen, ob ein bestimmtes Objekt im Bild sichtbar bzw. unsichtbar ist. Die Klasse `ACAlign` (`ActionConditionAlign`) überprüft, ob sich zwei Objekte „berühren“.

7.2 Eigenschaften des Systems

Die intelligente Steuerung eines Kamerasystems soll die Möglichkeiten eines automatischen Kamerasystems erweitern, um so größt-möglichen Nutzen für den Betreiber zu generieren. Die Schlüsseleigenschaft für ein solches System stellt die verbale Kommunikationsmöglichkeit dar. So sollen im Einzelnen Joystick-Interaktionsmöglichkeiten, das Speichern von Positionen, das Laden und Anfahren von Positionen und Einstellungen sowie die Möglichkeit der zeitwährenden Aktionen und bildfokussierende Fahrten möglich sein, um die Erwartungen des potentiellen Anwenders und die damit einhergehende Flexibilität zu erfüllen.

7.2.1 Joystick-Aktionen

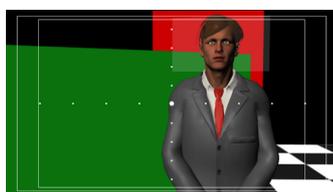
Automatische Kamerasysteme bieten Joystick-Interaktion als systemeigene Funktionalität an. Ein intelligentes Steuersystem soll diese Aktionen verbal verfügbar machen und dabei die reine Richtungsvorgabe durch sinnvolle Erweiterungen ergänzen, sodass das System leicht durch sprachgebräuchliche Kommandos geführt werden kann.

Das Kamerasystem kann mit dem vorgestellten Satz an Kommandos in jede beliebige Richtung bewegt werden. Mit Hilfe der Joystick-Aktionen können zusätzlich bestimmte Bedingungen im Bild gefordert werden. So kann zum Beispiel die Kamera so lange verfahren werden, bis bestimmte Objekte im Bild sichtbar bzw. unsichtbar sind. Ferner können bestimmte Objekte an bestimmte Positionen im Bild montiert werden.

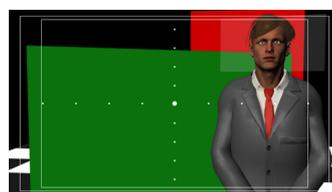
Abbildung 7.4 (a) zeigt die Ausgangssituation für eine konditionale Joystick-Anweisung (vgl. Kapitel 5.4). Das Bild zeigt eine *Halbnahe-Einstellung* bei der auf der linken Seite ein blaues Objekt in das Bild hinein ragt. Abbildung 7.4 (b) zeigt das Ergebnis nach Ausführung der Korrektur-Anweisung *Fahre zurück, bis das das blaue Objekt nicht mehr zu sehen ist*. In Abbildung 7.4 (c) wurde der Moderator durch Schwenken der Kamera an Safe-Title angelehnt. Der Ausgangspunkt für diese Anweisung war ebenfalls Abbildung 7.4 (a).



(a) Ausgangssituation - das blaue Objekt ist im Bild zu sehen



(b) Situation, nach dem die Kamera so weit nach links verfahren wurde, dass das blaue Objekt nicht mehr im Bild zu sehen ist



(c) Der Moderator wurde an Safe-Action angelehnt

Abbildung 7.4: Unterschiedliche Endpositionen für studiotypische Joystick-Aktionen. Das Regelsystem berücksichtigt die geforderten Bedingungen für das Beenden einer Joystick-Aktion.

Aus Kapitel 4 geht außerdem hervor, dass ein intelligentes Kamerasystem eine Unterscheidung unbestimmter Quantitäten vollzieht. Abbildung 7.5 zeigt die unterschiedlichen Laufweiten in Abhängigkeit der unterschiedlichen Quantitätsangaben.

Abbildung 7.5 (a) demonstriert die Ausgangssituation. (b) zeigt einen horizontalen Schwenk mit verringerter, (c) mit normaler und (d) mit erweiterter Quantität. Klar zu erkennen ist, dass die Laufweiten des Systems in Abhängigkeit der Quantität unterschiedlich berücksichtigt

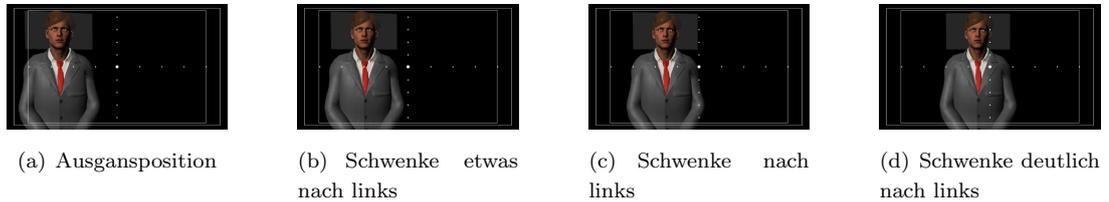


Abbildung 7.5: Abweichungen für unterschiedlich quantifizierte Schwenks nach links. Das Steuersystem bewegt die Kamera in Abhängigkeit der übergebenen Quantität.

werden. Für eine optimale Verwendbarkeit müssen diese Laufweiten immer konstant sein, sodass sich ein Benutzer an die jeweilige Laufweite gewöhnen kann und die gewünschte Distanz im Vorhinein abzuschätzen weiß. Die genannte Eigenschaft wird insbesondere durch die bildbasierte Regelung des Joysticks erfüllt.

In der Phase der *Einstellungsfindung* ist es üblich, Einstellungen zu erproben, um dann, wenn die Einstellung nicht die gewünschte Wirkung hat, auf die vorherige Einstellungen zurück gehen zu können. Der Joystick eines intelligenten Kamerasystems muss in der Lage sein, ein Zurück-Kommando richtig zu interpretieren. In dem hier vorgestellten System wird *zurück* deshalb als Richtung interpretiert und ausgeführt. In Abbildung 7.6 (a) ist die Ausgangssituation dargestellt. Darstellung (b) zeigt das Ergebnis eines Kamera-Schwenks nach links. (c) zeigt die Endsituation, nachdem das System *zurück*-beordert wurde. Die Kamera hat dabei nach rechts geschwenkt. In Abbildung 7.6 (d) folgt ein erneutes, *vermindertes zurück*.

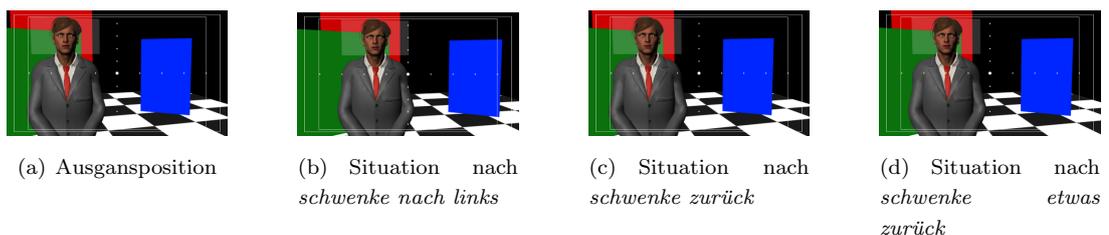


Abbildung 7.6: *Zurück* als Eigenschaft des Joysticks. Das Steuersystem berücksichtigt quantifizierte Zurück-Anweisungen. Das System fährt bei *fahre zurück* zurück zur Ausgangsposition. Wird ein *vermindertes zurück* gefordert, fährt das System weniger zurück als ohne Quantitätsangabe.

Die Kamera schwenkt daraufhin wieder etwas nach links. Der Moderator befindet sich nun zwischen den Positionen aus den Abbildungen (b) und (c). In dieser Bildreihe wird der Vorteil von Safe-Aktion und Safe-Title klar. Die Distanzen lassen sich hier leichter erkennen, als wenn keine vertikalen Grenzen im Bild zu sehen wären.

7.2.2 Laden und Speichern

Das System muss in der Lage sein sowohl zuvor gespeicherte Positionen als auch Standard-Einstellungen anzufahren. Zuvor gespeicherte Positionen müssen dabei auf den aktuellen Moderator angepasst werden können. Abbildung 7.7 zeigt das Ergebnis der Anpassung.



(a) Einstellung mit einem 176cm großen Moderator.



(b) Einstellung mit einem 184cm großen Moderator.



(c) Einstellung mit einem 184cm großen Moderator und angewandter Größenkompensation.

Abbildung 7.7: Vergleich von Einstellungen für Moderatoren unterschiedlicher Größe. In Bild (b) wird der Kopf des Moderators angeschnitten. Diese Situation sollte das System vermeiden. Bild (c) zeigt die automatisch korrigierte Einstellung in diesem Szenario.

In Abbildung 7.7 (a) ist die Position, wie sie vom Benutzer gespeichert wurde, dargestellt. In dieser Einstellung ist der Moderator 176cm groß. (b) zeigt die gleiche Einstellung mit einem 184cm großen Moderator. In der Abbildung ist gut zu erkennen, dass er für diese Einstellung zu groß ist und somit sein Kopf an der oberen Bildkante angeschnitten wird. In Abbildung 7.7 (c) ist die modifizierte Einstellung mit dem 184cm großen Moderator zu sehen. Es gleicht, wie gefordert, dem in Abbildung 7.7 (a) gezeigtem Bild.

Das System unterstützt außerdem die Verwendung von Standard-Einstellungen. Die Ausgestaltung der nötigen Parameter für Nachrichtensendungen wurde in Kapitel 3 anhand der einschaltquotenstärksten Nachrichtensendungen vorgenommen. Abbildung 7.8 zeigt eine Totale Einstellung im verwendeten Test-Set.



Abbildung 7.8: Beispiel einer *Totale-Einstellung* von HULICO.

Das Test-Set ist bewusst einfach gehalten, erfüllt aber dennoch die Eigenschaften eines sendefähigen Sets. Die dargestellten Farben dienen nur der Unterscheidung der einzelnen Objekte und haben sonst keine Bedeutung. In der dargestellten Abbildung zeigt die Totale alle (relevanten) Elemente des Sets. In der Abbildung sind das der Moderator, die virtuelle Präsentationsfläche (grün) sowie ein Objekt (blau). Die rote Fläche soll nicht berücksichtigt werden. Die Einstellung entspricht damit den Eigenschaften einer *Totalen-Einstellung*. Abbildung 7.9 zeigt einen Vergleich mit der, bei der Nachrichtensendung *RTL Aktuell* gezeigten, *Amerikanischen-Einstellung* und einer mit HULICO angefahrenen (rechten) *Amerikanischen-Einstellung* mit den gleichen Parametern, wie sie in Kapitel 3 festgestellt wurden.



(a) *Amerikanische* bei *RTL Aktuell*



(b) *Amerikanische* bei HULICO

Abbildung 7.9: Vergleich der *Amerikanische-Einstellungen* von *RTL Aktuell* und HULICO.

Die Ähnlichkeit der beiden Darstellungen ist unbestritten. Abbildung 7.10 zeigt einen Vergleich der *Halbnahe-* und *Nahe-Einstellungen* zwischen HULICO und den realen Nachrichten. Die jeweils gefundenen Parameter wurden in HULICO entsprechend übernommen.

7.2.3 Zeitwährende Kommandos

Zeitwährende Kommandos erlauben es dem Kamerasystem bestimmte, längerfristige Aufträge zu erteilen. Eine solcher Auftrag könnte lauten: *Halte den Moderator mittig*. Während der Aktivität eines zeitwährenden Kommandos können weitere Kommandos an das System übergeben werden. Da die bildbasierte Regelung des zeitwährenden Kommandos mit dem neu übergebenen Kommando in Konflikt geraten könnte, werden alle Fahrplanweisungen, die während einer zeitwährenden Aktion an das System übergeben werden, gerätebasiert ausgeführt (vgl. Kapitel 6.4.1). Es hat sich gezeigt, dass die bildbasierte Regelung des Systems zu unempfindlich ist, um eine solche Aktion auf Sendung zu geben. Für die *Einstellungsfindung* ist diese Funktion jedoch verwendbar.

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO



(a) *Halbnahe* bei RTL Aktuell



(b) *Halbnahe* bei HULICO



(c) *Nahe* bei RTL Aktuell



(d) *Nahe* bei HULICO

Abbildung 7.10: Vergleich der *Halbnahe-Einstellung* und *Nahe-Einstellung* von RTL Aktuell und HULICO.

7.2.4 Bildfokussierende Fahrten

Im Nachrichten- und Magazinbereich werden oft Fahrten von Einstellungen auf andere Einstellungen durchgeführt. Während der Fahrt sollen bestimmte Eigenschaften des Bilds berücksichtigt werden. Insbesondere gilt dies für den Abstand zwischen Bildoberkante und dem Kopf des Moderators (*Luft*). Dieser Abstand muss konstant gehalten werden. Bei einer Fahrt von z.B. einer *Halbnahen* auf eine *Nahe*, wird der Fixpunkt während der Fahrt stets eingehalten und somit diese Anforderung erfüllt. Abbildung 7.11 zeigt den Verlauf einer Fahrt von einer *Halbnahen* (a) auf eine *Nahe-Einstellung* (c).

Die Zwischenposition (b) wurde während der Fahrt aufgenommen. Gut zu erkennen ist der gleichbleibende Abstand (*Luft*) über dem Kopf des Moderators. Dieser Umstand kann allerdings nicht über ein visuell-basiertes System erreicht werden. An dieser Stelle wurde das Kamerasystem entsprechend erweitert, um bei Zoomfahrten automatisch den Tilt mitbewegen zu können. Durch eine generierte Linsendatei können die Öffnungswinkel der Linse für bestimmte Zoomstellungen ermittelt und dann vom Tilt kompensiert werden. Während einer Zoomfahrt fährt so der Tilt mit.



Abbildung 7.11: Fixpunkt während einer Zoomfahrt. Der Abstand zwischen der Oberkante des Kopfes und dem oberen Bildrand, bleibt während der Fahrt konstant.

Das HULICO-System ist in der Lage, einfache Fahrten zwischen Standard-Einstellungen durchzuführen. Allerdings hat sich gezeigt, dass die rein visuelle Regelung für komplexe Fahrten so nicht ausreicht. Während der Fahrt werden Fehler zu spät erkannt und das führt dazu, dass die Fahrten für den Zuschauer nicht mehr zielgerichtet wirken.

7.2.5 Flexibilität des Prototyps

Flexibilität wird im Kontext dieser Arbeit in zwei Ausprägungen betrachtet. Zum einen unterscheidet sich das Verständnis über die Ausgestaltung der Standard-Einstellungen zwischen den einzelnen Anwendungen und zum anderen muss sich das System geänderten Situationen während einer Sendung anpassen können.

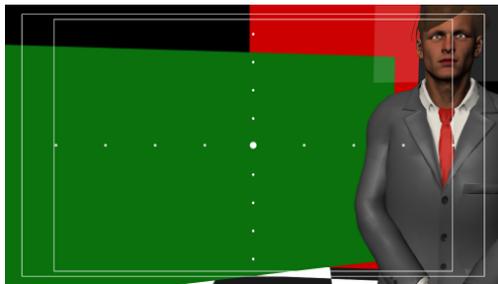
Das System ist höchst-flexibel verwendbar. Die Architektur erlaubt eine freie Wahl der Sprachverarbeitung und das hierarchisch organisierte Handlungsmodell eine jederzeit gute Anpassbarkeit an die lokalen Studio-Bedingungen. So können die einzelnen Einstellungen anhand von typischen Eigenschaften, wie sie in der Film- und Fernsehbranche üblich sind, definiert werden. Die Formulierung durch Regeln erlaubt es, das Verhalten des Systems schnell und leicht anzupassen und in das System zu integrieren.

Durch die Berücksichtigung aller relevanter Parameter, die auch einem Kameramann zur Verfügung stehen, kann das System auf kleinste Änderungen in der Welt, im aufgenommenen Bild sowie an der kameratragenden Kinematik reagieren. Das ist immer dann nötig, wenn der Moderator größer oder kleiner ist als der, der zur Festlegung der Einstellungen verwendet wurde, oder wenn er sich während der Sendung nicht auf der richtigen Position im Studio befindet. Abbildung 7.12 zeigt die Adaptivität des Systems auf geänderte Rahmenbedingungen durch Anwendung der Kompensationsstrategie (vgl. 4.3.2).

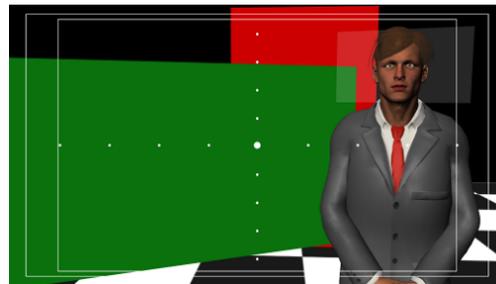
7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO



(a) Ausgangseinstellung mit einem 176cm großen Moderator.



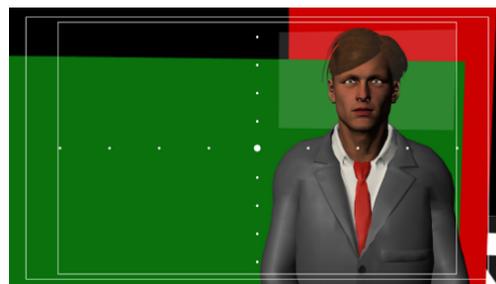
(b) Ausgangseinstellung mit einem 184cm großen Moderator, der +20cm neben der Moderationsposition steht. Die Kompensation ist deaktiviert.



(c) Ausgangseinstellung mit einem 184cm großen Moderator, der +20cm neben der Moderationsposition steht. Nach erfolgter Kompensation.



(d) Ausgangseinstellung mit einem 184cm großen Moderator, der -20cm neben der Moderationsposition steht. Die Kompensation ist deaktiviert.



(e) Ausgangseinstellung mit einem 184cm großen Moderator, der -20cm neben der Moderationsposition steht. Nach erfolgter Kompensation.

Abbildung 7.12: Vergleich der Einstellungen bei unvorhergesehenen Änderungen an der Szene. Die Bildfolge zeigt unterschiedliche Positionen und Größen des Moderators bei der Anfahrt einer bestimmten Einstellung mit und ohne Kompensation des hier vorgestellten Systems.

In Abbildung 7.12 (a) ist die Ausgangseinstellung dargestellt. Diese zeigt den Moderator vor einer Präsentationsfläche (grün). Der Moderator ist in dieser Einstellung 176cm groß. Abbildung 7.12 (b) zeigt die gleiche Einstellung mit einem Moderator, der aus Sicht der Kamera 20cm zu weit rechts steht. Dieser Moderator ist 184cm groß. Das Bild zeigt den Moderator angeschnitten, die Bild-Kompensation ist deaktiviert. Abbildung 7.12 (c) zeigt die gleiche Situation mit aktivierter Kompensation. Die Präsentationsfläche wirkt durch die Änderungen im Bild etwas mehr in die Tiefe hin orientiert. Die Einstellung ist etwas größer. In Abbildung 7.12 (d) befindet sich der Moderator aus Sicht der Kamera um 20cm zu weit links. Der Moderator ist weiterhin 184cm groß. Die Kamera ist ohne Kompensation auf die gewünschte Position gefahren, deshalb ist der Moderator angeschnitten dargestellt. Das Bild wirkt unharmonisch. Der starken Information auf der linken Seite des Bildes kann der mittig stehende Moderator nicht entgegen wirken. Abbildung 7.12 (e) zeigt diese Einstellung mit Kompensation des Bildes. Es ist nur noch ein Teil der Präsentationsfläche sichtbar, allerdings ist das Gleichgewicht im Bild wieder hergestellt.

7.3 Evaluation des Prototyps im Rahmen einer Benutzerstudie

Die objektive Bewertung eines Systems hinsichtlich der vollständigen verbalen Steuerbarkeit sowie hinsichtlich des richtigen Verhaltens in einer Studio-Umgebung fällt schwer, weil sie nutzerabhängig und nicht messbar ist. Eine Evaluation unter Regie-Personal soll deshalb diese Frage beantworten. Anhand der Benutzerstudie soll festgestellt werden, ob das in dieser Arbeit vorgestellte intelligente Steuerungssystem die Verwendung von automatischen Kamerasystemen vereinfacht und ob die vom System generierten Aktionen menschenähnlich ausgeführt werden.

In den folgenden Kapiteln werden die Durchführung der Studie sowie deren Ergebnisse vorgestellt. In Kapitel 7.3.1 wird der genaue Ablauf der Befragung dargelegt. Daran schließt sich in Kapitel 7.3.2 die Präsentation der Probandengruppe sowie in Kapitel 7.3.3 die Vorstellung des verwendeten Fragebogens an. Die Hypothese der Evaluation ist in Kapitel 7.3.4 beschrieben. Auf die Ergebnisse der Befragung in Kapitel 7.3.5 schließt sich eine Diskussion der Ergebnisse in Kapitel 7.3.6 an.

7.3.1 Ablauf der Befragung

Die Befragung wurde jeweils in einem Büro am Arbeitsplatz der jeweiligen Probanden durchgeführt. Für die Probanden der Fernsehsender ZDF und CenterTV wurden dazu die Pausen zwischen den Sendungen genutzt. Die anderen Probanden wurden in ihren Büros besucht und befragt. Zur Befragung wurde das Testsystem in Form von zwei Laptops aufgebaut. Auf einem Laptop lief das VizRt-Render System zur Simulation des Kamerabildes und für den Blick auf die Welt. Auf dem anderen Laptop lief das RoboKam[©]-Core-System sowie der entwickelte Prototyp HULICO. Die Simulation zeigt eine einfache Szene, wie sie bereits auf den bisher gezeigten Bildern zu sehen ist (vgl. Abbildung 7.4). Es existiert ein Moderator, eine Präsentationsfläche (grün) sowie ein „Objekt“ (blau). Die rot gefärbte Fläche stellt einen weiteren Kontrast in der Szene dar. Dieses Objekt wird jedoch nicht vom System überwacht.

Die Probanden wurden begrüßt und darauf hingewiesen, dass sie an einer akademischen Evaluation einer intelligenten, verbalen Steuerung für ein automatisches Kamerasystem im Bereich Nachrichtensendungen und Magazine teilnehmen. Außerdem wurde darauf hingewiesen, dass in dieser Ausbaustufe der Fokus auf der verbalen Steuerbarkeit sowie auf den erreichten Endpositionen des Systems liegt und dass das System zwar verbal steuerbar ist, aber keine phonetische Überführung des gesprochenen Wortes im Prototypen verwendet wird. Stattdessen geschieht die Eingabe von Kommandos beim Prototypen textuell, um so fehlerhafte Interpretationen der Kommandos, auf Grund von undeutlicher Aussprache, zu vermeiden.

Daraufhin wurden die Komponenten des Systems kurz vorgestellt und dann mit der Erklärung der einzelnen möglichen Kommandos begonnen. Es wurde darauf hingewiesen, dass für die zukünftige Verwendung mehrerer automatischer Kamerasysteme alle Kommandos in der Form [ACTOR] [CMD] aufgebaut sind, um so ein bestimmtes Kamerasystem ansprechen zu können. Danach wurden die einzelnen Kommandos in der Reihenfolge aus Kapitel 5 vorgestellt und jeweils vorgeführt. Bei Joystick-Kommandos wurde auf die Möglichkeit der Quantitäten hingewiesen.

Konditional-Kommandos wurden anhand eines in der Szene enthaltenen blauen Objekts vorgestellt. Hier wurden die Kommandos *Kamera1 schwenke nach links bis das Objekt nicht mehr sichtbar ist* bzw. *Kamera1 schwenke nach rechts bis das Objekt sichtbar ist* vorgeführt. Bei manchen Probanden wurde weiterhin noch *Kamera1 fahre nach links bis der Moderator an SA anliegt* (SA steht hier für Safe-Action) vorgestellt. Diese Probanden hatten zuvor nachgefragt, ob sich Konditional-Kommandos nur auf die Sichtbarkeit beziehen. Diese Frage wurde dann verneint und als weiteres Beispiel oben genannte Äußerung angeführt. Die Einstellungsadaption

von zuvor gespeicherten Positionen wurde mit Hilfe einer Positions- bzw. einer Größenänderung des Moderators im Render-System erklärt. Die Präsentation der Kommandos endete mit der Vorstellung von zeitwährenden Kommandos. Hier wurde ein zeitwährendes Kommando angestoßen und dann der virtuelle Moderator im Set verschoben. Zuletzt wurde darauf hingewiesen, dass es bei den Joystick-Kommandos eine Bedeutungskollision für den Begriff *zurück* gibt und wie dieser Begriff im Prototypen behandelt wird (siehe dazu Kapitel 5.2.1). Außerdem wurde darauf hingewiesen, dass die hinterlegten Einstellungsgrößen von den Sendungen *tagesschau*, *heute* und *RTL Aktuell* (vgl. Kapitel 3.3.2) stammen und dass diese leicht auf die lokalen Interpretationen der Begriffe *Nahe*, *Halbnahe* etc. änderbar sind. Nun bestand die Möglichkeit selbst das System zu bedienen. Schließlich wurde der Fragebogen auf Papier ausgefüllt. Alle Probanden haben unentgeltlich an der Befragung teilgenommen.

7.3.2 Teilnehmer der Befragung

Für die Evaluation wurden insgesamt 13 unterschiedliche Berufsträger von unterschiedlichen Sendeanstalten sowie freie Regie-Mitarbeiter und beratend tätiges Personal befragt. Unter den Probanden befanden sich Regisseure, die für das Zweite Deutsche Fernsehen (ZDF) tätig sind, Mediengestalter mit regieführenden Aufgaben und erfahrene Auszubildende des Senders CenterTV in Köln. Außerdem wurden ein Student mit Studioerfahrung sowie eine wissenschaftliche Mitarbeiterin mit Regieerfahrung von der Fakultät für Medien- und Elektrotechnik der FH Köln sowie Mitarbeiter der RTLeaders GmbH und der Avirtico GmbH - Agentur für Echtzeitgrafik befragt. Die befragten Personen sind zwischen 22 und 55 Jahre alt und haben alle bereits mehrjährige Erfahrung in der Arbeit mit manuellen oder automatischen Kamerasystemen. Es haben zwölf männliche und ein weiblicher Proband an der Befragung teilgenommen.

7.3.3 Verwendeter Fragebogen

Der verwendete Fragebogen besteht aus zwei Teilen. Der erste Teil lehnt sich an den Standardfragebogen ISONORM 9241/10 [Prü93] an und bewertet das Verhalten des Systems sowie die Auswahl und Umsetzung der einzelnen Kommandos. Dieser Teil des Fragebogens enthält insgesamt 16 Multiple-Choice-Fragen (ISO-1 bis ISO-16), von denen 14 jeweils vier Antwortoptionen und jeweils zwei Fragen eine Freitext-Antwort bieten. Die Antwortoptionen für die 14 Fragen entsprechen einer 4-stufigen Likert-Skala [Lik32]. Abbildung 7.13 zeigt diese Optionen mit Angabe der vergebenen Punkte. Der Fragebogen ist vollständig im Anhang zu finden.

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

- : ich stimme gar nicht zu (1 Punkt)
- : ich bin unentschlossen, mit negativer Tendenz (2 Punkte)
- + : ich bin unentschlossen, mit positiver Tendenz (3 Punkte)
- ++ : ich stimme vollkommen zu (4 Punkte)

Abbildung 7.13: Antwort-Möglichkeiten für Teil 1 des Fragebogens.

Frage ISO-1 *Die Software ist unkompliziert zu steuern* dient der Bewertung der Bedienerfreundlichkeit. Nachdem die Probanden die natürlichsprachliche Kommunikation mit einem menschlichen Kameramann gewohnt sind, zeigt diese Frage, ob das System prinzipiell die Sprache im Studio verwendet.

Frage ISO-2 *Im Vergleich zur bestehenden Software erleichtert HULICO die Steuerbarkeit des Roboters* zeigt den direkten Vergleich mit einem automatischen Kamerasystem. Die Probanden können an dieser Stelle erklären, ob sie dieser Aussage zustimmen oder nicht. Mit Hilfe eines Freitext-Feldes soll erklärt werden, wieso sie sich für die jeweilige Antwort entschieden haben.

Frage ISO-3 *Die Software bietet alle Funktionen, um die anfallenden Aufgaben effizient zu erledigen* sowie Frage ISO-4 *Die Software ist gut auf die Anforderungen meiner Arbeit zugeschnitten* dienen der Bestimmung, ob der Prototyp das ganze Spektrum von nötigen Kommandos abdeckt oder nicht. Frage ISO-3 bietet die Möglichkeit, fehlende Funktionen in Form eines Freitexts festzuhalten.

Die Fragen ISO-5 *Die angebotenen Kommandos kenne ich*, ISO-6 *Die angebotenen Kommandos verwende ich (häufig)*, ISO-7 *Die angebotenen Kommandos reichen mir zum Einrichten einer Position* und ISO-8 *Die angebotenen Kommandos reichen mir zum Fahren einer Sendung* zeigen die Vollständigkeit der Sprachschnittstelle. Sie sollen feststellen, ob die verwendeten Kommandos allgemein bekannt und individuell verwendet werden. Außerdem zeigen diese Fragen, ob die ausgewählten Kommandos zum Einrichten einer Sendung sowie zum Fahren einer Sendung reichen. Frage ISO-7 und Frage ISO-8 bieten die Möglichkeit, evtl. fehlende Funktionen anzugeben.

Frage ISO-9 *Die Kommandos verhalten sich so, wie erwartet* evaluiert allgemein, ob das Verhalten des Systems dem erwarteten Verhalten entspricht. Frage ISO-10 *Die Joystick-Kommandos werden wie erwartet ausgeführt* bezieht sich speziell auf das Verhalten der Joystick-Kommandos. Frage ISO-11 *Der Höhenausgleich funktioniert, wie erwartet* und Frage ISO-12 *Den Höhenaus-*

7.3 Evaluation des Prototyps im Rahmen einer Benutzerstudie

gleich könnte man für eine Sendung verwenden soll feststellen, ob die Reaktion des Systems auf Änderungen in der Umwelt richtig reagiert. Sollte dies nicht der Fall sein, hat der Proband die Möglichkeit, wahrgenommene Auffälligkeiten in Form von Freitext festzuhalten.

Die Fragen ISO-13 *Um das System zu verwenden, müsste ich mich nicht umgewöhnen*, ISO-14 *Um das System zu verwenden, brauche ich nicht lange, um es verwenden zu können* und ISO-15 *Um das System zu verwenden, kann ich mich auf das Bild konzentrieren. Systemdetails werden abstrahiert* zeigen die Nutzbarkeit des Prototyps. Hier soll untersucht werden, wie leicht die Gewöhnung an das System vollzogen werden kann. Eine schnelle Gewöhnung erleichtert die Integrierbarkeit des Systems in eine Studioumgebung.

Frage ISO-16 *Welche weiteren Funktionen würden Sie sich für das System wünschen?* bietet dem Probanden die Möglichkeit, weitere wünschenswerte Funktionen für das System anzugeben.

Der zweite Teil des Fragebogens besteht aus einer deutschen Übersetzung¹ des System Usability Scales (SUS) [Bro96] und soll evaluieren, inwieweit die Idee als Ganzes in der Praxis verwendbar ist. Die einzelnen Fragen werden im Folgenden mit SUS-1 bis SUS-10 bezeichnet. Der SUS wurde angewandt, weil dieser bereits wissenschaftlich analysiert [Lew12, BFL09, BKM08] und außerdem sehr kurz ist. So kann er ergänzend zu den bereits vorgestellten Fragen verwendet werden.

Der SUS steht unter Lizenzvorbehalt und erfordert einen Hinweis auf den Urheber des SUS. Dieser Hinweis sei hier mit Verweis auf das Literaturverzeichnis erfolgt. Der SUS erlaubt eine allgemeine Aussage über die Verwendbarkeit einer Software für ein bestimmtes Problem. Die Probanden müssen zehn Fragen beantworten. Abbildung 7.14 zeigt die deutsche Übersetzung dieser 10 Fragen.

Für jede Antwort können zwischen 1 und 5 Punkte vergeben werden. 1 Punkt steht dabei für *Ich stimme nicht zu*, 5 Punkte stehen für *Ich stimme zu*. Es müssen bei diesem Fragebogen alle zehn Fragen beantwortet werden. Sollte man eine Frage nicht beantworten können, sollen drei Punkte vergeben werden. Die erreichte Punktzahl wird auf einen Endschlüssel (Scale) umgerechnet. Dieser bildet das Ergebnis auf einen Wert zwischen 0 und 100 ab und gibt Aufschluss darüber, wie gut sich das System benutzen lässt.

7.3.4 Hypothesen der Benutzerstudie

Das System wurde ausgehend von theoretischen und praktischen Erfahrungen entwickelt. Deshalb ist davon auszugehen, dass es die prinzipiellen Anforderungen erfüllt. Hinsichtlich der

¹<http://minds.coremedia.com/2013/09/18/sus-scale-an-improved-german-translation-questionnaire/>

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

SUS-1 Ich denke, dass ich dieses System gerne regelmäßig nutzen würde.

SUS-2 Ich fand das System unnötig komplex.

SUS-3 Ich denke, das System war leicht zu benutzen.

SUS-4 Ich denke, ich würde die Unterstützung einer fachkundigen Person benötigen, um das System benutzen zu können.

SUS-5 Ich fand, die verschiedenen Funktionen des Systems waren gut integriert.

SUS-6 Ich halte das System für zu inkonsistent.

SUS-7 Ich glaube, dass die meisten Menschen sehr schnell lernen würden, mit dem System umzugehen.

SUS-8 Ich fand das System sehr umständlich zu benutzen.

SUS-9 Ich fühlte mich bei der Nutzung des Systems sehr sicher.

SUS-10 Ich musste viele Dinge lernen, bevor ich mit dem System arbeiten konnte.

Abbildung 7.14: Deutsche Übersetzung der Fragen des *System Usability Scales* (SUS).

berücksichtigten Kommandos ist zu erwarten, dass diese für die Steuerung einer Kamera ausreichen. Dennoch ist die verwendete Sprache sehr vielfältig, sodass bestimmte Begriffe fehlen könnten.

Das Verhalten des Systems ähnelt dem eines Kameramanns. Deshalb wird erwartet, dass die meisten Aktionen ebenfalls Zustimmung finden. Fahrten lagen nicht im Fokus des Systems, sodass hier Schwächen festgestellt werden dürften.

7.3.5 Ergebnisse der Befragung

In diesem Kapitel werden die Ergebnisse der Befragung vorgestellt. Für jede Frage wird dabei die Anzahl der jeweiligen Bewertungen sowie der Durchschnittswert und die Standardabweichung angegeben. Der Durchschnittswert und die Standardabweichung sind anhand einer Punkteverteilung von einem bis vier Punkten berechnet. Die negative Antwort (-) zählt dabei ein Punkt und eine positive Antwort (++) vier Punkte. Die jeweiligen Tendenzen (-/+) gehen entsprechend mit zwei bzw. drei Punkten in die Bewertung ein. Tabelle 7.1 stellt die Ergebnisse von Teil 1 des Fragebogens dar.

Für Frage ISO-1 *Die Software ist nicht kompliziert zu steuern* haben fünf Probanden voll

7.3 Evaluation des Prototyps im Rahmen einer Benutzerstudie

Tabelle 7.1: Ergebnisse aus dem ersten Teil der Befragung. Es ist jeweils die Frage sowie die Anzahl der jeweiligen Antworten sowie der Durchschnitt und die Standardabweichung angegeben. Zu jeder Antwort steht in Klammern die vergebene Punktzahl: z.B. ++ (4). Für Frage ISO-2 wurde nur gezählt, ob die Antwort positiv oder negativ ausgefallen ist. Frage ISO-16 ist eine Freitextantwort und fehlt deshalb.

Frage	-- (1)	- (2)	+ (3)	++ (4)	Durchschnitt	Standardabweichung
ISO-1	0	1	7	5	3.3	0.63
ISO-2	2	–	–	10	–	–
ISO-3	0	0	7	6	3.5	0.52
ISO-4	0	2	4	7	3.4	0.77
ISO-5	0	0	1	12	3.9	0.28
ISO-6	0	1	0	12	3.8	0.55
ISO-7	0	1	2	10	3.7	0.63
ISO-8	2	1	1	9	3.3	1.18
ISO-9	0	0	2	11	3.8	0.38
ISO-10	0	0	2	11	3.8	0.38
ISO-11	0	0	2	11	3.8	0.38
ISO-12	1	1	4	7	3.3	0.95
ISO-13	1	1	6	5	3.2	0.90
ISO-14	1	0	5	7	3.4	0.87
ISO-15	0	0	4	9	3.7	0.48

zugestimmt (++). Sieben Probanden haben eine positive Tendenz (+) und eine Person eine negative Tendenz (-) bescheinigt. Auf Nachfrage gab dieser an, seine Kameraleute nicht verbal zu korrigieren. Aus seiner Sicht können die präsentierten Bilder eines Kameramanns immer gesendet werden. Andere Probanden gaben an, die in HULICO verwendeten Kommandos zwar zu kennen, allerdings sind sie z.T. andere Begrifflichkeiten innerhalb der Kommandos gewöhnt und müssten sich deshalb erst in die vorgegebenen Kommandos einarbeiten bzw. sich die vorgestellten Kommandos erst angewöhnen. Bei einem realen Kameramann schätzen sie die Flexibilität der verwendbaren Anweisungen. Die Durchschnittsbewertung liegt bei 3.3, die Standardabweichung bei 0.63.

Bei Frage ISO-2 *Im Vergleich zum bestehenden System erleichtert HULICO die Steuerbarkeit* gaben zehn Probanden an, dass HULICO die Steuerung vereinfacht (++), zwei sehen eine kompliziertere Steuerung (-) und ein Proband hat sich der Stimme enthalten. Unter den positiven Bewertungen wurden Vorteile wie *bildbezogene Steuerung, natürlichsprachliche, schnellere Kommunikation, die Vorzüge eines automatischen Vorschlags für eine spezielle Einstellung*, die

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

intuitive Steuerung und damit das *Verhindern von Missverständnissen bei der Anweisung* und die Möglichkeit *parallel den Bildmischer bedienen zu können* aufgeführt. Die negativen Antworten führen die *unmittelbare Steuerung mit Hilfe von Joysticks* sowie den *erhöhten Aufwand* bei der Kontrolle der Kamera an.

Bei Frage ISO-3 *Die Software bietet alle Funktionen um die anfallenden Aufgaben effizient zu erledigen* stimmten sechs Probanden dieser Aussage voll zu (++) . Sieben haben eine positive Tendenz (+) bei dieser Aussage. Als Bemerkung wurde angegeben, dass *potenziell in bisher nicht betrachteten Szenarien Funktionen fehlen könnten*, dass *eine absolute Bildbeschreibung hilfreich wäre*, *manche Fahrten nicht sendefähig sind*, *ein manuelles Eingreifen nicht möglich ist*, *Dialogsituationen nicht berücksichtigt sind* und dass eine *automatische Adaption an ein von einer anderen Kamera gesendetes Bild* vorteilhaft wäre.

Frage ISO-4 *Die Software ist gut auf die Anforderungen meiner Arbeit zugeschnitten* haben zwei Probanden mit negativer Tendenz (-), vier mit positiver Tendenz (+) und sieben positiv (++) beantwortet. Es ergibt sich ein Durchschnitt von 3.4 und eine Standardabweichung von 0.77.

Die Frage ISO-5 *Die angebotenen Kommandos kenne ich* haben 12 Probanden positiv (++) beantwortet und ein Proband mit positiver Tendenz (+). Der Durchschnitt liegt bei 3.9, die Standardabweichung bei 0.28.

Frage ISO-6 *Die angebotenen Kommandos verwende ich* wurde ebenfalls von 12 Probanden positiv (++) und von einem mit negativer Tendenz (-) beantwortet. Der Durchschnitt liegt bei 3.8, die Standardabweichung bei 0.55.

Frage ISO-7 *Die angebotenen Kommandos reichen mir zum Einrichten einer Einstellung* beantwortete ein Proband mit negativer Tendenz (-), 2 mit positiver Tendenz (+) und zehn positiv (++) . Der Durchschnitt beträgt 3.7. Die Standardabweichung liegt bei 0.63.

Zwei Probanden haben Frage ISO-8 *Die angebotenen Kommandos reichen mir zum Fahren einer Sendung* negativ (- -) beantwortet, ein Proband mit negativer Tendenz (-) und ein weiterer mit positiver Tendenz (++) . Neun Probanden haben diese Frage positiv beantwortet (++) . Die Probanden gaben an, dass *die Verwendung in einer Sendung explizit getestet werden sollte*, *Referenzen auf beliebige Eigenschaften von Objekten (z.B. Ellbogen des Moderators) sowie Funktionen zum Überschreiben der Parameter fehlen*, *Fahrten nicht explizit zur Verfügung stehen*, *eine generelle Abneigung für die Verwendung eines „intelligenten Systems“ für eine Sendung besteht*, dass *der Proband nicht für Nachrichten arbeitet*, *zeitliche Angaben für die Fahrten auf Positionen fehlen* und dass *Zweier bzw. Dreier- Einstellungen nicht berücksichtigt sind*. Durchschnittlich liegt die Bewertung bei 3.3. Die Standardabweichung liegt bei 1.18.

7.3 Evaluation des Prototyps im Rahmen einer Benutzerstudie

Bei Frage ISO-9 *Die Kommandos verhalten sich so, wie erwartet* stimmten elf Probanden dieser Aussage ohne Einschränkung zu (++). Zwei Probanden waren unentschlossen mit positiver Tendenz (+). Der Durchschnitt liegt bei 3.8 mit einer Standardabweichung von 0.38.

Elf Probanden stimmten auch der Aussage aus Frage ISO-10 *Die Joystick-Kommandos werden, wie erwartet ausgeführt* zu (++). Zwei Probanden hatten eine positive Tendenz (+) und gaben folgende Bemerkungen an: *Die Kopffreiheit sollte bei Vertikalbewegungen beibehalten werden* und *es sollte möglich sein, gleichzeitig mehrere Achsen zu fahren*. Der Durchschnitt bei dieser Frage liegt bei 3.8, die Standardabweichung bei 0.38.

Für Frage ISO-11 *Positionsadaption wird wie erwartet durchgeführt* gaben zwei Probanden eine positive Tendenz (+) an, elf stimmten vollständig positiv (++). Der Durchschnitt liegt bei 3.8, die Standardabweichung bei 0.38.

Frage ISO-12 *Die Positionsadaption könnte man für eine Sendung verwenden* beantworteten sieben Probanden positiv (++), vier mit positiver Tendenz (+), einer mit negativer Tendenz (-) und ein Proband vollständig negativ (- -). Als Begründung für die Bewertung gaben die Probanden an, dass *das Resultat der Adaption jeweils vorher mit dem aktuellen Moderator getestet werden sollte*, dass *Positionen mit mittel-großen Moderatoren aufgenommen werden sollten, der Ausgleich nur im Off verwendbar ist, für den Ausgleich eine feste Untergrenze für die Kamerahöhe festgelegt werden muss* und dass *die Fahrt auf die Einstellungen nicht sendefähig ist*. Der Durchschnitt liegt bei 3.3, die Standardabweichung bei 0.95.

Ein Proband beantwortete Frage ISO-13 *Um das System verwenden zu können müsste ich mich nicht stark umgewöhnen* negativ (- -), einer mit negativer Tendenz (-), sechs mit positiver Tendenz (+) und fünf vollständig positiv (++). Der Durchschnitt liegt bei 3.2. Die Standardabweichung bei 0.90.

Auf die Frage ISO-14 *Um das System verwenden zu können brauche ich nicht lange* antworteten sieben Probanden positiv (++), fünf mit positiver Tendenz (+) und einer negativ (- -). Der Durchschnitt liegt bei dieser Frage bei 3.4. Die Standardabweichung bei 0.87.

Frage ISO-15 *Um das System verwenden zu können, muss ich nicht viele Systemdetails berücksichtigen* beantworteten vier Probanden mit positiver Tendenz (+) und sieben positiv (++). Der Durchschnitt liegt bei 3.4 mit einer Standardabweichung von 0.87.

Bei Frage ISO-16 *Welche weiteren Funktionen würden Sie sich für das System wünschen* gaben die Probanden an, dass *Fahrten mit der Angabe von Zeiten fehlen, eine Möglichkeit bestehen sollte, den Bezug zwischen Vordergrund und Hintergrund zu definieren* sowie dass *weitere Begrifflichkeiten* sowie „Jargon“ ergänzt werden sollte.

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

In Teil 2 des Evaluationsbogens wurde die allgemeine Usability des Systems erfragt. Die Tabelle 7.2 stellt die Ergebnisse von Teil 2 des Fragebogens zusammen.

Tabelle 7.2: Ergebnisse aus dem zweiten Teil der Befragung. Es ist jeweils die Frage sowie die Anzahl der jeweiligen Antworten sowie der Durchschnitt und die Standardabweichung angegeben. Zu jeder Antwort steht in Klammern die von SUS vergebene Punktzahl: z.B. ++ (5).

Frage	-- (1)	- (2)	+ - (3)	+ (4)	++ (5)	Durchschnitt	Standardabweichung
SUS-1	0	0	4	3	6	4.2	0.89
SUS-2	8	4	1	0	0	1.5	0.66
SUS-3	0	0	0	5	8	4.6	0.51
SUS-4	9	2	0	1	1	1.7	1.3
SUS-5	0	0	1	3	9	4.6	0.65
SUS-6	9	1	3	0	0	1.5	0.88
SUS-7	0	0	3	2	8	4.4	0.87
SUS-8	10	1	2	0	0	1.4	0.77
SUS-9	0	0	2	6	5	4.2	0.72
SUS-10	8	2	1	2	0	1.8	1.16

Frage SUS-1 *Ich denke, dass ich das System gerne regelmäßig nutzen würde* haben sechs Probanden mit fünf Punkten, drei mit vier Punkten und vier Probanden mit drei Punkten bewertet. Im Durchschnitt ergibt das eine Bewertung von 4.2 Punkten. Die Standardabweichung liegt bei 0.89.

Frage SUS-2 *Ich fand das System unnötig komplex* wurde von acht Probanden mit einem Punkt, von vier mit zwei Punkten und von einem Probanden mit drei Punkten versehen. Im Schnitt ergibt das 1.5 Punkte und eine Standardabweichung von 0.66.

Frage SUS-3 *Ich denke das System war leicht zu benutzen* wurde von acht Probanden mit fünf Punkten bewertet und von fünf Probanden mit vier Punkten. Im Durchschnitt ergibt das eine Bewertung von 4.6 Punkte und eine Standardabweichung von 0.51.

Die Frage SUS-4 *Ich denke ich würde die Unterstützung einer fachkundigen Person benötigen, um das System benutzen zu können* beantworteten neun Probanden mit einem Punkt, zwei mit zwei Punkten und jeweils ein Proband mit vier und einer mit fünf Punkten. Im Durchschnitt ergibt das 1.7 Punkte und eine Standardabweichung von 1.3.

Frage SUS-5 *Ich fand die verschiedenen Funktionen des Systems waren gut integriert* wurde von neun Probanden mit fünf Punkten, von drei Probanden mit vier und von einem Probanden mit drei Punkten bewertet. Im Durchschnitt ergibt das 4.6 Punkte und eine Standardabweichung von 0.65.

7.3 Evaluation des Prototyps im Rahmen einer Benutzerstudie

Frage SUS-6 *Ich halte das System für zu inkonsistent* beantworteten neun Probanden mit einem Punkt, einer mit zwei Punkten und drei Probanden mit drei Punkten. Der Durchschnitt dieser Frage liegt bei 1.5, die Standardabweichung liegt bei 0.88.

Frage SUS-7 *Ich glaube, dass die meisten Menschen sehr schnell lernen würden, mit dem System umzugehen* beantworteten acht Probanden mit fünf Punkten, zwei mit vier Punkten und drei mit drei Punkten. Der Durchschnitt dieser Frage liegt bei 4.4, die Standardabweichung bei 0.87.

Frage SUS-8 *Ich fand das System sehr umständlich zu benutzen* wurde von zehn Probanden mit einem Punkt, von einem Proband mit zwei und von drei Probanden mit drei Punkten beantwortet. Der Durchschnitt liegt bei 1.4 Punkten und die Standardabweichung bei 0.77.

Frage SUS-9 *Ich fühlte mich bei der Nutzung des Systems sehr sicher* beantworteten fünf Probanden mit fünf Punkten, sechs mit vier Punkten und zwei Probanden mit drei Punkten. Der Durchschnitt liegt bei 4.2. Die Standardabweichung bei 0.72.

Frage SUS-10 *Ich musste viele Dinge lernen, bevor ich mit dem System arbeiten konnte* beantworteten acht Probanden mit einem Punkt, zwei Probanden mit zwei Punkten, einer mit drei Punkten und zwei mit vier Punkten. Der Punkte-Durchschnitt dieser Frage liegt bei 1.8. Die Standardabweichung bei 1.16.

Im Durchschnitt ergibt die Bewertung durch den Fragebogen einen SUS-Scale von 85.4 Punkten.

7.3.6 Diskussion

Zusammengefasst zeigt die Befragung, dass das System, so wie es hier vorgestellt wurde, alle Anforderungen erfüllt, soweit es im vorgesehenen Arbeitsumfeld der Einrichtungsfindung verwendet wird. Im Folgenden werden die Ergebnisse in Boxplots vorgestellt und daraufhin einige der Antworten interpretiert und diskutiert. Die Fragen ISO-2 und ISO-16 werden dabei zuerst behandelt, da sie Freitextantworten sind und deshalb in den folgenden Boxplots nicht berücksichtigt werden konnten.

7.3.6.1 Diskussion der Ergebnisse - Teil 1

Frage ISO-2 *Im Vergleich zur bestehenden Software erleichtert HULICO die Steuerbarkeit des Roboters* zeigt, dass die Güte des System vom gewohnten Umfeld der Probanden abhängt. Steuern diese die vorhandenen automatischen Systeme ausschließlich selbst während der Sendung und sind es nicht gewohnt ausgebildete Kameraleute anzuweisen, fällt die Gewöhnung an ein verbal steuerbares System schwer. Gleiches gilt, wenn viel oder ausschließlich mit menschlichen

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

Kameraleuten gearbeitet wird. Hier können Kommandos in beliebiger Form gegeben werden. Diese Flexibilität der menschlichen Kameraleute kann der vorgestellte Prototyp (noch) nicht leisten. Dennoch zeigt die Auswertung der Frage, dass für die meisten Probanden die verbale Interaktion mit dem System durchaus ein Gewinn in der Arbeit mit einer automatischen Kamera darstellt. Gerade für Probanden, die sowohl mit automatischen Systemen als auch mit menschlichen Kameraleuten arbeiten, ist das durch HULICO implementierte System interessant. HULICO erlaubt es hier eine (weitgehend) gleichbleibende Interaktionsform mit der Kamera beizubehalten.

Frage ISO-16 *Welche weiteren Funktionen würden Sie sich für das System wünschen?* gibt den Probanden die Gelegenheit fehlende Funktion anzugeben. Die Probanden nannten hier, dass manche Fahrten nur unbefriedigend ausgeführt werden und dass *Jargon* berücksichtigt werden sollte. Außerdem sollte eine Möglichkeit bestehen, den Bezug zwischen Vordergrund und Hintergrund zu definieren.

Die Ausführung der Fahrten ist tatsächlich eine Schwachstelle des verwendeten Prototyps. Während einer Fahrt werden entstehende Fehler im Bild, wie z.B. eine zunehmende Kopffreiheit, zu spät erkannt, um rechtzeitig ohne visuelle Folgen kompensiert werden zu können. Die träge Verbindung zwischen Tracking und HULICO könnte dafür ein Grund sein. Eine weitere Ursache könnte die zu langsame Kompensationsgeschwindigkeit des Prototyps sein. Beides sind Punkte die für eine Weiterentwicklung untersucht werden sollten. Die Berücksichtigung von „Jargon“ stellt kein Problem dar und kann, dank der verwendeten Grammatik, jederzeit ins System integriert werden. Allerdings muss dafür eine Untersuchung vorausgehen, welche Ausprägung der Umgangssprache im jeweiligen Studio verwendet wird. Da dieser Wunsch immer nur speziell für einen Probanden passen würde, wurde im Prototypen nur die vollständige Form der Kommandos berücksichtigt. Die verbalen Formulierungen für die Beschreibung der Beziehung zwischen Vorder- und Hintergrund wurde im Rahmen der Voruntersuchungen nicht genannt und fehlen deshalb im System. Dieser Punkt wurde nur von einem Probanden gewünscht. Die anderen Probanden haben einen solchen Aspekt nicht erwähnt. Eine nähere Untersuchung müsste die Aktionen für diese Funktion sowie deren verbale Formulierung klären. Abbildung 7.15 zeigt die Ergebnisse der Befragung anhand von sogenannten Kastengrafiken (BoxPlots).

Der Vorteil dieser Darstellung ist, dass die Meinung der Probanden bereinigt von Ausreißern dargestellt wird. Jede Frage aus Teil 1 ist in Abbildung 7.15 entlang der X-Achse durch die Bezeichner I-1 - I-15 markiert. Die Fragen aus Teil 2 sind in Abbildung 7.16 mit den Bezeichnern S-1 - S-10 angegeben. Die Boxen in diesen Grafiken geben den Bereich an, den die mittleren

7.3 Evaluation des Prototyps im Rahmen einer Benutzerstudie

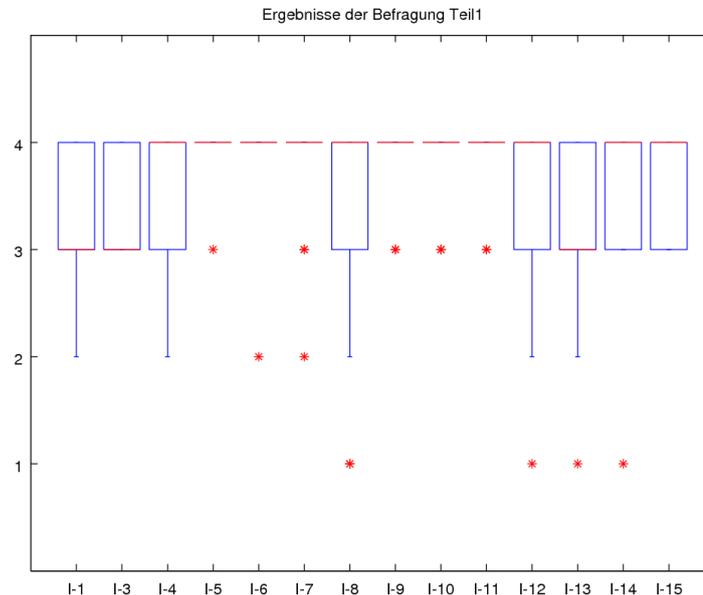


Abbildung 7.15: Auswertung Teil 1 des Evaluationsbogens. Entlang der X-Achse sind die Fragen ISO-1 bis ISO-15 durch I-1 bis I-15 dargestellt. Die Fragen ISO-2 sowie ISO-16 des Fragebogens sind keine Multiple-Choice-Fragen und fehlen in der Grafik. Entlang der Y-Achse bedeutet 1 keine Zustimmung zur jeweiligen Aussage, 4 bedeutet volle Zustimmung. Die einzelnen Mediane (rote Linie) zeigen, dass alle Fragen positiv bewertet wurden.

50%¹ der Befragten auf die jeweilige Frage geantwortet haben [Tuk77]. Dieser wird durch die blaue Box im Grafen dargestellt. Der rote Strich innerhalb dieser Boxen zeigt den Median innerhalb der relevanten Antworten. Die vertikalen Striche außerhalb der Boxen werden als Antennen bezeichnet. Sie geben an, in welchem Bereich außerhalb der Box die weiteren Antworten liegen. Die Länge der Antennen ist auf das 1,5-fache der Boxbreite beschränkt. Werte, die außerhalb der Antennen liegen, werden als Ausreißer bezeichnet und sind in den Diagrammen durch Sterne dargestellt. Entlang der Y-Achse entspricht der Wert 1 einer negativen Bewertung im Fragebogen und 4 einer positiven Bewertung. Die Werte 2 bzw. 3 werden entsprechend zugeordnet. In Abbildung 7.15 fehlen die Fragen ISO-2 und ISO-16. Diese Fragen sind Freitext-Antworten und können nicht in der Grafik dargestellt werden.

Die Darstellung 7.15 zeigt, dass der Prototyp durchweg positiv angenommen worden ist. Die bereinigte Meinung konzentriert sich auf die positive Seite (3 bzw. 4 Punkte) der möglichen

¹Von allen Antworten werden die besten 25% der Bewertungen und die schlechtesten 25% der Bewertungen entfernt.

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

Antworten. Die meisten Mediane liegen bei vollständiger Zustimmung zur jeweiligen Aussage.

Die Mediane für die Fragen ISO-1 und ISO-3 liegen (nur) bei einer positiven Tendenz. Dieses Ergebnis kann darin begründet liegen, dass der vorgestellte Prototyp aufgrund der weggelassenen Spracherkennung so bewertet wurde. Die Suche nach bestimmten Kommandos hat oft viel Zeit gekostet und so eine spontane Bedienung des Systems verhindert. Gerade in Bezug auf die vollständig positive Bewertung der enthaltenen Kommandos sowie der daraufhin ausgeführten Aktionen, überrascht dieses Ergebnis.

Ein eindeutiges Ergebnis bringen die Fragen ISO-5 bis ISO-7. Bei diesen Fragen geht es um den angebotenen Satz an Kommandos. Das Ergebnis zeigt, dass die dem System bekannten Kommandos auch den Probanden durchweg bekannt sind und verwendet werden. Außerdem reichen sie, um eine Position einzurichten. Bei der Frage, ob die Kommandos zum Abfahren einer Sendung reichen, ist die Meinung dagegen nicht mehr so unumstößlich. An dieser Stelle sei aber auf die Tragweite einer solchen Aussage hingewiesen. In einer Sendung müssen auch unvorhergesehene Ereignisse in Betracht gezogen werden, die menschliche Kameramänner leicht berücksichtigen können. Für die Bewertung eines Systems hinsichtlich des Einsatzes während einer Sendung fehlte den meisten Probanden vermutlich Erfahrung mit dem System. Auch sei hier angemerkt, dass manche Fahrten nicht so sauber ausgeführt werden, wie das für eine Verwendung in einer Sendung nötig wäre. Gerade in Havarie-Szenarien könnte eine fehlende Fahrt eine Sendung ruinieren. Der Satz an Kommandos zur verbalen Steuerung eines Kamerasystems enthält neben Stopp- und Joystick-Kommandos auch objektbezogene und Konditional-Kommandos. Außerdem berücksichtigt der Prototyp Kommandos zum Laden und Speichern von Einstellungen sowie Kommandos, um zeitwährende Aktionen anzustoßen. Die Evaluation der Fragen ISO-5 bis ISO-7 zeigt, dass die angebotenen Kommandos allen Probanden geläufig sind, verwendet werden und zum Einrichten einer Einstellung ausreichen.

Wiederum sehr positiv ist das Verhalten des Systems aufgenommen worden (ISO-9). Die ausgeführten Aktionen der Kommandos allgemein sowie die speziellen Aktionen bei Joystick-Kommandos (ISO-10), haben die Probanden überzeugt. Auch die Positionsadaption wird richtig und verwendbar durchgeführt (ISO-11 und ISO-12). Schwächen hat hier das System nur bei der Anfahrt der Positionen gezeigt. Die Fahrt ist aufgrund des *Schwimmens* und *Pumpens* nicht sendefähig. Fahrten lagen jedoch nicht im Fokus des Systems. Für dieses System steht eine überzeugende Endposition im Zentrum der Betrachtung.

Die Fragen ISO-13 *Um das System verwenden zu können müsste ich mich stark umgewöhnen*, ISO-14 *... brauche ich lange* sowie Frage ISO-15 *... muss ich viele Systemdetails berücksichtigen*

zeigen im Graphen ebenfalls durchwegs positive Antworten.

7.3.6.2 Diskussion der Ergebnisse - Teil 2

Das Gesamtergebnis des SUS ist genau dann positiv, wenn die Antworten alternierend positiv und negativ, beginnend mit einer positiven Antwort gegeben werden. Abbildung 7.16 zeigt das Ergebnis des Teils 2 aus der Befragung in einem Boxplot.

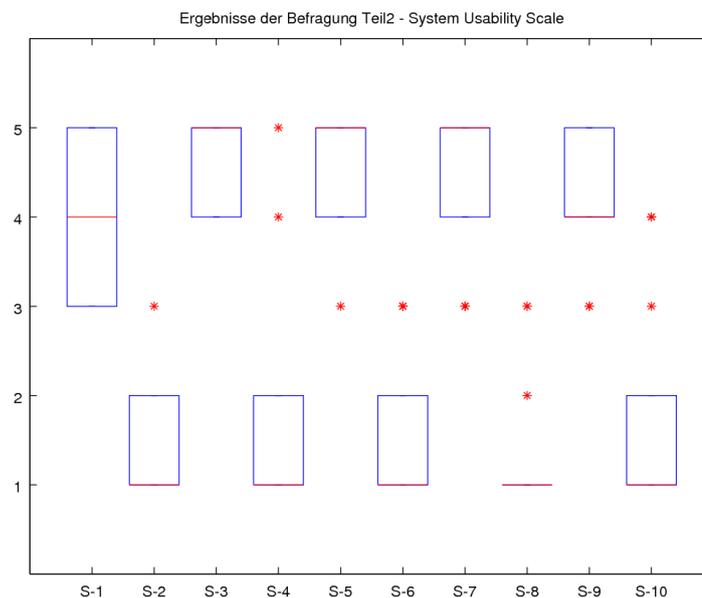


Abbildung 7.16: Auswertung Teil 2 des Fragebogens. Die Fragen SUS-1 bis SUS-10 sind durch S-1 bis S-10 bezeichnet. Für eine 100%-positive Bewertung müssen die ungeraden Fragen (1,3,5. . .) mit 5 Punkten und die geraden Fragen (2,4,6 etc.) mit 1 Punkt beantwortet werden. Der Median (rote Linie) zeigt bereits das positive Ergebnis der Befragung. Die Sterne stellen Ausreißer unter den Ergebnissen dar.

Im Boxplot zeigt sich, dass der Median der Antworten meist auf 1 bzw. 5 Punkten liegt und sich somit das exzellente Abschneiden des Prototypen auch im Boxplot zeigt. Die Ausreißer bei Frage SUS-4 könnten in der Tatsache liegen, dass diese Probanden die Frage primär auf die Einführung eines solchen Systems bezogen haben. Zur Einführung muss das System von einer fachkundigen Person vorgestellt werden, um die Möglichkeiten voll ausschöpfen zu können. Die anderen Probanden erwarten vermutlich ebenfalls eine solche Einführung in das System, fühlen sich dann aber in der Lage das System selbstständig anzuweisen.

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

Bei Frage SUS-9 liegt der Boxplot-Median bei 4 Punkten. Diese Frage ist zusammen mit Frage SUS-1 die einzige, bei der von einer klaren Zustimmung bzw. Abneigung abgewichen wird. Dieser Umstand mag in der Tatsache liegen, dass das System neu und unbekannt und deshalb schwer einzuschätzen ist.

Das Gesamt-Ergebnis der SUS-Evaluation ist mit 85.4 Punkten definitiv positiv zu sehen. Nach Sauro und Lewis [SL12] kann man die erreichten Punkte in das amerikanische Notensystem umrechnen. Abbildung 7.17 zeigt die Projektion des SUS-Scale auf das amerikanische Notensystem. Bis 51.7 Punkte entspricht das Ergebnis der Note F. Bis 62.6 Punkte entspricht der Scale der Note D, bis 72.5 Punkten einem C, bis 78.8 Punkten einem B. Alles darüber entspricht der Note A [SL12]. Durch den Bezug auf das Notensystem ist die Güte des Systems leicht nachvollziehbar. Der erreichte Wert von 85.4 liegt in der Abbildung klar im Bereich von A und stellt damit eindeutig eine zukunftssträchtige Entwicklung dar.

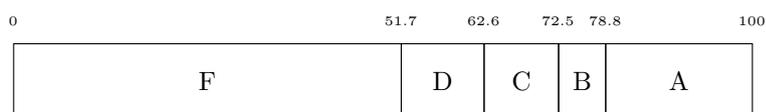


Abbildung 7.17: Abbildung des SUS-Scales in das amerikanische Notensystem von F bis A (nach [SL12]).

Das in dieser Arbeit vorgestellte System entspricht den in Kapitel 4 formulierten Anforderungen. Es werden sowohl Joystick-Aktionen als auch Aktionen zum Laden und Speichern von Positionen sowie zeitwährende Aktionen unterstützt. Außerdem unterstützt das, in Form des Prototyps HULICO, entwickelte System einfache Fahrten zwischen Positionen, wie sie auch in Nachrichtensendungen Verwendung finden.

Im Rahmen der Joystick-Aktionen erlaubt das System sowohl die Verwendung von Quantitäten als auch die Formulierung von Konditional-Anweisungen. Im Rahmen der durchgeführten Evaluation wurde festgestellt, dass die Joystick-Anweisungen wie erwartet ausgeführt werden.

Beim Speichern und Laden speichert das System neben den Positionsinformationen auch die aktuelle Bildgestaltung ab. Dies erleichtert es, Verfehlungen, die durch falsch positionierte Moderatoren bzw. durch unterschiedlich große Moderatoren hervorgerufen werden, automatisch zu kompensieren. Die Evaluation des Systems hat ergeben, dass die Kompensation einer Position durchaus sendefähige Einstellungen hervorbringt. Die Anfahrt auf diese Einstellungen ist jedoch nicht sendefähig, weil während der Fahrt Objekte im Bild zu schwimmen und pumpen

7.3 Evaluation des Prototyps im Rahmen einer Benutzerstudie

beginnen. Für das Ziel dieser Arbeit steht jedoch das Erreichen sendefähiger Einstellungen im Vordergrund, sodass die problematische Anfahrt auf eine Einstellung nicht ins Gewicht fällt.

Neben zuvor gespeicherten Einstellungen, kennt das System auch Standard-Einstellungen, wie sie in der Fernsehwelt geläufig sind. Die Evaluation hat ergeben, dass den bereits bekannten Einstellungen nur die *Zweier-* bzw. *Dreier-Einstellung* fehlt. Prinzipiell entsprechen die vorgeführten Standard-Einstellungen jedoch dem allgemeinen Verständnis. Die leichte Anpassbarkeit der Vorgabe wurde ebenfalls positiv angenommen.

Das System erlaubt es, Kommandos über längere Zeit in der Abarbeitung zu halten. Die Vorteile einer solchen Funktion für die Verwendung in Nachrichtensendungen ist unbestritten, um die Kamera bei kleinen Bewegungen des Moderators ausgerichtet zu halten. Für die komplexere Verwendung im Rahmen der Einstellungsfindung sollten die Achsen des Systems jedoch synchronisiert werden, sodass ein Perspektivenwechsel mit gleichbleibender Bildkadrierung nicht erst beim Übergang in den Stillstand durchgeführt wird. Das System verliert während der Fahrt die Kadrierung und nimmt sie vor Stillstand wieder auf. Die Durchführung dieser Aktion auf Basis von „visuellen Eindrücken“ scheint nicht verwendbar zu sein. Eine Verbesserung könnte hier die Steuerung der Orientierung der Kamera anhand eines Zielpunktes bringen. Dennoch können die zeitwährenden Kommandos während einer Sendung eine bestimmte Einstellung halten. Nur so erklärt sich die durchwegs gute Bewertung der Aktionen in der Evaluation.

Im Rahmen der Evaluation hat sich die Bedeutung der Flexibilität gezeigt. Fast alle Probanden haben sich mindestens eine spezielle Änderung des Setups bzw. des Systems an sich gewünscht. Beginnend bei der Definition der Standard-Einstellungen, über die Festlegung der maximalen Fahrdistanzen oder der Geschwindigkeiten, wurden fast alle Parameter des Systems auf Änderbarkeit abgefragt. HULICO basiert auf einer Regelmaschine, die eine schnelle und leichte Änderbarkeit garantiert.

Die Beschreibung von Fahrten wurde von einigen Probanden gewünscht. Da das zugrundeliegende, visuelle Regelsystem nur einfache Fahrten abbilden kann, würde eine Ergänzung in dieser Richtung dem System noch mehr Zuspruch entgegen bringen. Da in den vorangegangenen Feldstudien allerdings zu wenige Informationen zu Fahrten gesammelt werden konnten und der Rahmen dieser Arbeit für eine Ergänzung nicht reicht, sei an dieser Stelle auf eine zukünftige Arbeit verwiesen. Mit dem Fokus auf Nachrichten und Magazinsendungen und trotz der Einschränkung auf einfache Fahrten, hat das System in der vollzogenen Evaluation eine exzellente Bewertung erreicht. Es berücksichtigt die gewohnten Kommandos in einer Studioumgebung und führt Aktionen wie erwartet aus.

7. IMPLEMENTIERUNG UND EVALUATION DES PROTOTYPS HULICO

Kapitel 8

Zusammenfassung und Ausblick

Automatische Kamerasysteme erfreuen sich wachsender Beliebtheit in Fernsehstudios vieler Nachrichten- und Magazinsendungen. Die nahtlose Integration dieser Systeme bereitet jedoch noch Probleme, weil einmal aufgenommene Positionen nicht situationsgerecht angepasst und das System nicht verbal angesprochen werden kann.

8.1 Das Ziel dieser Arbeit

Die vorliegende Arbeit zielte darauf ab, grundlegende Eigenschaften der Kameraführung für Nachrichten- und Magazinsendungen zu identifizieren. Die Erkenntnisse sollten dazu führen, automatische Systeme so steuern zu können, dass sie sich wie menschengeführte Systeme anweisen und damit leicht in eine etablierte Produktionsumgebung eingliedern lassen.

8.2 Die Inhalte dieser Arbeit

Durch Befragung von Fachpersonal und Analyse von Fernsehsendungen wurde festgestellt, dass die wesentliche Eigenschaft einer intelligenten Kamerasteuerung im Fernsehstudio die verbale Instruierbarkeit ist. Die damit verbundene Literaturrecherche zeigte, dass die bisher auf diesem Gebiet entwickelten Systeme diesen Gedanken noch nicht aufgegriffen haben. Auf dieser Erkenntnis aufbauend wurde ein Satz von sieben Anweisungstypen identifiziert, mit deren Hilfe ein Kamerasystem eingestellt und über eine Sendung hinweg gesteuert werden kann.

Im Rahmen der vorgestellten Anweisungstypen werden besondere Begrifflichkeiten verwendet, die es erlauben Verfehlungen im Bild einfach zu formulieren. Das nötige Vokabular wird ebenfalls vorgestellt und erklärt.

8. ZUSAMMENFASSUNG UND AUSBLICK

Im Rahmen der Befragungen des Fachpersonals und der Analyse der Fernsehsendungen wurde deutlich, dass typische Qualitäten eines Bildes verlangt werden, um die Wiedererkennbarkeit eines Formats zu unterstützen. Sie sind sendungs-spezifisch definiert und werden bei jeder Produktion beachtet. Die Identifikation dieser wesentlichen Bildeigenschaften erlaubt es, eine automatische Kompensationsstrategie für spontan geänderte Studiobedingungen zu entwickeln. Sie trägt dafür Sorge, dass die Kameraposition automatisch immer so variiert werden kann, dass der Bildausschnitt in allen wichtigen Punkten dem Sendungsformat entspricht.

8.3 Das Ergebnis dieser Arbeit

Die Ergebnisse wurden in einem Prototyp integriert. Dieser ist in der Lage verbal ausgedrückte Anweisungen zu verstehen und in Steuerkommandos umzusetzen. Dabei konnten viele der umgangssprachlichen Variationen der Anweisungen in der entwickelten Grammatik berücksichtigt werden.

Die Architektur des Prototyps trägt der Flexibilität der Fachbegriffe Rechnung, sodass sein Verhalten schnell auf die Bedürfnisse einzelner Anwender angepasst werden kann.

Der Prototyp ist in der Lage, die gewünschten Einstellungen an die aktuellen Studiobedingungen anzupassen und sender- bzw. studiospezifische Vorgaben zu berücksichtigen.

Bei der Evaluation des Systems anhand des Prototyps hat sich gezeigt, dass der gewählte Ansatz die Anforderungen im Fernsehstudio, im Rahmen der *Einstellungsfindung* voll und für die *Sendekontrolle* zum Teil erfüllt. Die befragten Berufsträger haben das System außerordentlich positiv bewertet und gaben an, dass sie alle im System enthaltenen Kommandos kennen und verwenden. Außerdem stellten sie fest, dass die angebotenen Kommandos zur verbalen Steuerung eines automatischen Kamerasystems ausreichen. Das Verhalten des Systems wurde ebenfalls durchweg als richtig und wie erwartet bewertet. Der in dieser Arbeit vorgestellte Prototyp kann deshalb als erster, wesentlicher Schritt hin zu einem intelligenten, automatischen Kamerasystem angesehen werden.

Für den Einsatz des Systems in einem Realszenario müssen die Informationen, die bisher von der Simulation zur Verfügung gestellt werden, in einer realen Studio-Umgebung bestimmt werden. Dies wird die zukünftige Aufgabe für die Weiterentwicklung des Systems sein. Wenn diese Informationen zur Verfügung stehen, kann das System leicht in einer realen Studioumgebung installiert werden.

8.3 Das Ergebnis dieser Arbeit

Für die zukünftige Erweiterung des Systems sollten die Sprachkommandos um Fahrtbeschreibungen ergänzt werden. Der Prototyp zeigt jedoch auch ohne diese Erweiterungen bereits, dass ein solches System gerade für kleine Sendeanstalten einen Mehrwert generiert.

8. ZUSAMMENFASSUNG UND AUSBLICK

Anhang A

Appendix

A. APPENDIX

Evaluation des HULICO

Angelehnt an ISONORM 9241/10

Name:

Berufsstand:

..... Jahre Erfahrung mit der Führung / Arbeit mit...

<input type="checkbox"/>	manueller/n,
<input type="checkbox"/>	automatischer/n Kamerasysteme(n): _____

Unterstützt die HULICO die Erledigung Ihrer Aufgaben, ohne Sie als Benutzer unnötig zu belasten?

1	Die Software...	--	-	+	++	Die Software...
	Ist kompliziert zu steuern					Ist unkompliziert zu steuern

2	Im Vergleich zur bestehenden Software erleichtert HULICO die Steuerbarkeit des Roboters.					
	Ja, weil					
	Nein, weil					

3	Die Software...	--	-	+	++	Die Software...
	Bietet nicht alle Funktionen, um die anfallenden Aufgaben effizient zu erledigen					Bietet alle Funktionen, um die anfallenden Aufgaben effizient zu erledigen
Folgende Funktionen fehlen mir:						

4	Die Software...	--	-	+	++	Die Software...
	...ist schlecht auf die Anforderungen meiner Arbeit zugeschnitten					...ist gut auf die Anforderungen meiner Arbeit zugeschnitten.

Abbildung A.1: Seite 1 des Evaluationsbogens

Die angebotenen Kommandos...		--	-	+	++	Die angebotenen Kommandos...
5	... kenne ich nicht					... kenne ich
6	... verwende ich nicht					... verwende ich (häufig)
7	... reichen mir nicht zum Einrichten einer Position					... reichen mir zum Einrichten einer Position
Folgende fehlen mir:						
8	... reichen mir nicht zum Fahren einer Sendung					... reichen mir zum Fahren einer Sendung
Folgende fehlen mir:						

Die Kommandos...		--	-	+	++	Die Kommandos...
9	... verhalten sich nicht so, wie erwartet.					... verhalten sich so, wie erwartet.
Folgendes ist mir aufgefallen:						

Die Joystick-Kommandos...		--	-	+	++	Die Joystick-Kommandos..
10	...werden nicht wie erwartet ausgeführt					...werden wie erwartet ausgeführt.
Folgendes ist mir aufgefallen:						

Abbildung A.2: Seite 2 des Evaluationsbogens

A. APPENDIX

	Der Höhenausgleich...	--	-	+	++	Der Höhenausgleich...
11	...wird nicht wie erwartet ausgeführt					...funktioniert, wie erwartet
12	... kann man so nicht verwenden					... könnte man für eine Sendung verwenden
Folgendes ist mir aufgefallen:						

	Um das System zu verwenden...	--	-	+	++	Um das System zu verwenden...
13	... müsste ich mich stark umgewöhnen					... müsste ich mich nicht umgewöhnen
14	... brauche ich lange um es verwenden zu können					... brauche ich nicht lange, um es verwenden zu können.
15	... muss ich viele Systemdetails berücksichtigen					... kann ich mich auf das Bild konzentrieren. Systemdetails werden abstrahiert.

16	Welche weiteren Funktionen würden Sie sich für das System wünschen?				

Abbildung A.3: Seite 3 des Evaluationsbogens

Evaluation von HULICO
auf Basis des System Usability Scales

	K e i n e				Z u s t i m m u n g
Ich denke, dass ich dieses System gerne regelmäßig nutzen würde.					
Ich fand das System unnötig komplex.					
Ich denke, das System war leicht zu benutzen.					
Ich denke, ich würde die Unterstützung einer fachkundigen Person benötigen, um das System benutzen zu können.					
Ich fand, die verschiedenen Funktionen des Systems waren gut integriert.					
Ich halte das System für zu inkonsistent.					
Ich glaube, dass die meisten Menschen sehr schnell lernen würden, mit dem System umzugehen.					
Ich fand das System sehr umständlich zu benutzen.					
Ich fühlte mich bei der Nutzung des Systems sehr sicher.					
Ich musste viele Dinge lernen, bevor ich mit dem System arbeiten konnte.					

Abbildung A.4: Seite 4 des Evaluationsbogens

A. APPENDIX

Literaturverzeichnis

- [Ajd67] Kazimierz Ajdukiewicz. Die syntaktische Konnexität. *Studia Philosophica*, 1:1–27, 1935; English Translation in McCall, 207-231 (1967). 24
- [Ari00] Daniel Arijon. *Grammatik der Filmsprache*. Zweitausendeins, ISBN: 3-86150-336-0, 1st edition, 2000. 39, 42, 44, 45
- [Arm97] Alan A. Armer. *Lehrbuch der Film- & Fernsehregie*. Zweitausendundeins, ISBN: 3-86150-165-1, 1997. 38, 39, 42, 43
- [BCPW07] J. Baldridge, S. Catterjee, A. Palmer, and B. Wing. DotCCG and VisCCG: Wiki and Programming Paradigms for Improved Grammar Engineering with OpenCCG. *Proceedings of the GEAF 2007*, 2007. 30
- [BFL09] Simone Borsci, Stefano Federici, and Marco Lauriola. On the dimensionality of the system usability scale: a test of alternative measurement models. *Cognitive processing*, 10(3):193–197, 2009. 149
- [BK02] Jason Baldridge and Geert-Jan M. Kruijff. Coupling ccg and hybrid logic dependency semantics. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, pages 319–326. Association for Computational Linguistics, Stroudsburg, PA, USA, 2002. URL <http://dx.doi.org/10.3115/1073083.1073137>. 30, 32
- [BK03] Jason Baldridge and Geert-Jan M. Kruijff. Multi-modal Combinatory Categorical Grammar. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1, EACL '03*, pages 211–218. Association for Computational Linguistics, Stroudsburg, PA, USA, 2003. ISBN 1-333-56789-0. URL <http://dx.doi.org/10.3115/1067807.1067836>. 24, 25

LITERATURVERZEICHNIS

- [BKFK06] Nicolas Burtnyk, Azam Khan, George Fitzmaurice, and Gordon Kurtenbach. Show-Motion - camera motion based 3d design review. *ACM Symposium on Interactive 3D Graphics and Games.*, 2006. 15
- [BKM08] Aaron Bangor, Philip T. Kortum, and James T. Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human-Computer Interaction*, 24(6): 574–594, 2008. 149
- [Bli88] J. Blinn. Where am I? What am I looking at? *IEEE Computer Graphics and Applications*, pages 76–81, 1988. 10
- [Boc11] Julian Bochberg. Das audiovisuelle Design in Fernsehnachrichten und deren Unterschiede. Seminararbeit, Universität Koblenz-Landau, 2011. 1, 2
- [Bro96] John Brooke. SUS - A quick and dirty usability scale. *Usability Evaluation in Industry*, pages 189–194, 1996. 149
- [Bro09] Paul Browne. *JBoss Drools Business Rules*. Packt Publishing, ISBN: 978-1847196064, 1. Edition edition, 2009. 135
- [BTM00] William H. Bares, Somying Thainimit, and Scott McDermott. A model for constrained-based camera planning. *AAAI*, 2000. 11
- [BVF⁺13] N. C. Buchs, F. Volonte, Pugin F., Toso C., Fusaglia M., K. Gavaghan, P. E. Majno, M. Peterhans, S. Weber, and P. Morel. Augmented environments for the targeting of hepatic lesions during image-guided robotic liver surgery. In *Journal of Surgical Research*, pages 825–831. Elsevier, 2013. 17
- [CEC⁺05] F. Corcione, D. Esposito, D. Cucurullo, A. Settembre, N. Miranda, F. Amato, F. Pirozzi, and P. Caiazzo. Advantages and limits of robot-assisted laparoscopic surgery. In *Surgical Endoscopy and other inventional Techniques*, pages 117–119. Springer Science+Business Media Inc. 2004, 2005. 16
- [CFPS05] Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332, 2005. ISSN 1570-7075. URL <http://dx.doi.org/10.1007/s11168-006-6327-9>. 32

- [CM11] David L. Chen and Raymond J. Mooney. Learning to interpret natural language navigation instructions from observations. *AAAI*, 2011. 20, 21
- [DGZ92] S. Drucker, T. Galyean, and D. Zeltzer. Cinema: A system for procedural camera movements. *Proc. 1992 Symposium on Interactive 3D Graphics, ACM Press*, -, 1992. 14
- [Dru94] Steven M. Drucker. *Intelligent Camera control for Graphical Environments*. PhD thesis, Massachusetts Institute of Technology, 1994. 10
- [Fau08] Werner Faulstich. *Grundkurs Fernsehanalyse*. Wilhelm Fink Verlag, ISBN: 978-3-7705-4683-1, 1. Auflage edition, 2008. 3, 57
- [Fei01] Andreas Feininger. *Andreas Feininger's GROSSE FOTOLEHRE*. Heyne Verlag, ISBN: 3-453-17975-7, 4. Auflage edition, 2001. 19
- [FGI⁺09] Mary Ellen Foster, Manuel Giuliani, Amy Isard, Colin Matheson, Jon Oberlander, and Alois Knoll. Evaluating description and reference strategies in a cooperative human-robot dialogue system. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI'09*, pages 1818–1823. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009. URL <http://dl.acm.org/citation.cfm?id=1661445.1661737>. 22
- [FHN⁺09] D. L. Fouler, T. Hu, T. Nadkarni, P. K. Allen, and N. J. Hogle. Initial trial of a stereoscopic, insertable, remotely controlled camera for minimal access surgery. In *Surgical Endoscopy and other inventional Techniques*, volume 24, pages 9–15. Springer Science+Business Media Inc. 2004, 2009. 17
- [Fre18] G. Frege. *Der Gedanke. Eine logische Untersuchung*. G. Patzig, 1976. Zuerst erschienen in: Beiträge zur Philos. des deutschen Idealismus 1918-1919, 2. ergänzte Auflage edition, 1918. 24
- [Fry10] Michael Frye. *Digitale Landschaftsfotografie: Fotografieren wie Ansel Adams und Co.* mitp, 2010. ISBN 978-3826658969. 19
- [Giu06] Manuel Giuliani. A basic system for interpretation of utterances in natural language, based on the combinatory categorial grammar. Master's thesis, Technische Universität München, 2006. 24

LITERATURVERZEICHNIS

- [Giu11] M. Giuliani. *Comparing Classical and Embodied Multimodal Fusion for Human-Robot Interaction*. PhD thesis, Chair of Robotic and Embedded Systems, Technical University of Munich, 2011. 22
- [GW92] M. Gleicher and A. Witkin. Through-the-lens camera control. *Computer Graphics*, 26:331–340, 1992. 10
- [HA05] Schleicher H. and Urban A. *Filme machen, Technik - Gestaltung - Kunst*. Zweitausendeins, 1. Auflage edition, 2005. ISBN 3-86150-700-5. 10, 38, 39, 41, 42, 43
- [Hel04] Ralf Helbing. *Ein erweitertes Kameramodell - Methoden und Werkzeuge für die dynamische Kamerasteuerung in interaktiven Systemen*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2004. 10
- [Heu91] Klaus von Heusinger. *Kategoriale Unifikationsgrammatiken*, Arbeitspapier 37. Technical report, Fachbereich Sprachwissenschaft, Universität Konstanz, 1991. 24
- [Hic10] Knut Hieckethier. *Einführung in die Medienwissenschaft*. J.B. Metzler, 2010. ISBN 978-3476023513. 1
- [HO00] Nicolas Halper and Patrick Olivier. Camplan: A camera planning agent. In *In Smart Graphics 2000 AAAI Spring Symposium*, pages 92–100. AAAI Press, 2000. 11
- [HP08] Miroslav Holada and Matrin Pelc. The robot voice-control system with interactive learning. *New Developments in Robotics Automation and Control*, 2008. 19, 20
- [Hus28] E. Husserl. *Logische Untersuchungen*. Verena Mayer, Akademie Verlag GmbH, 28. ISBN 978-3-05-004391-3. 24
- [HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 10, 120
- [Ins10] Institut für Rundfunktechnik (IRT). *Technische Richtlinien zur Herstellung von Fernsehproduktionen für ARD, ZDF, ORF*, 2010. 43
- [JN08] Richard Johansson and Pierre Nugues. Dependency-based syntactic-semantic analysis with propbank and nombank. In *Proceedings of the Twelfth Conference on*

- Computational Natural Language Learning*, CoNLL '08, pages 183–187. Association for Computational Linguistics, Stroudsburg, PA, USA, 2008. ISBN 978-1-905593-48-4. URL <http://dl.acm.org/citation.cfm?id=1596324.1596355>. 22
- [Kan03] Pierre Kandorfer. *Lehrbuch der Filmgestaltung*. mediabook Verlag, 6. Auflage edition, 2003. ISBN 3-932972-18-X. 43
- [Kat98] Steven D. Katz. *Die richtige Einstellung - Shot by Shot. Zur Bildsprache des Films*. Zweitausendeins, 1. Auflage edition, 1998. ISBN 3861502291. 44, 45
- [KGD⁺03] A. Krupa, J. Gangloff, C. Doignon, M.F. de Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescaux. Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing. *Robotics and Automation, IEEE Transactions on*, 19(5):842–853, 2003. ISSN 1042-296X. 16
- [KKKL05] Seong-Young Ko, J. Kim, Dong-Soo Kwon, and Woo-Jung Lee. Intelligent interaction between surgeon and laparoscopic assistant robot system. *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pages 60–65, 2005. 16
- [KLK⁺04] Jonathan Kim, Yun-Ju Lee, Seong-Young Ko, Dong-Soo Kwon, and Woo-Jung Lee. Compact camera assistant robot for minimally invasive surgery: Kalar. *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 3:2587–2592, 2004. 18
- [KM06] Rohit J. Kate and Raymond J. Mooney. Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 913–920. Association for Computational Linguistics, Stroudsburg, PA, USA, 2006. URL <http://dx.doi.org/10.3115/1220175.1220290>. 21
- [Knu64] Donald E. Knuth. Backus normal form vs. backus naur form. *Commun. ACM*, 7(12):735–736, December 1964. ISSN 0001-0782. URL <http://doi.acm.org/10.1145/355588.365140>. 74
- [KRP⁺13] B. W. King, L. A. Reisner, A. K. Pandya, A. M. Composto, R. D. Ellis, and M. D. Klein. Towards an autonomous robot for camera control during laparoscopic

LITERATURVERZEICHNIS

- surgery. *Journal of Laparoendoscopic and Advanced Surgical Techniques*, pages 1027–1030, 2013. 17
- [Kru01] Geert-Jan M. Kruijff. *A Categorical Modal Architecture of Informativity: Dependency Grammar Logic & Information Structure*. PhD thesis, Charles University, Prague, Czech Republic, 2001. 32
- [KSJ⁺10] Myung-Jin Kim, Tae-Hoon Song, Seung-Hun Jin, Soon-Mook Jung, Gi-Hoon Go, Key-Ho Kwon, and Jae-Wook Jeon. Automatically available photographer robot for controlling composition and taking pictures. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 6010–6015, oct. 2010. ISSN 2153-0858. 19
- [KTRN10] T. Kollar, S. Tellex, D. Roy, and Roy N. Toward understanding natural language directions. *Proceedings of the ACM/IEEE, International Conference on Human-Robot Interaction*, pages 259–266, 2010. 21
- [Kuc05] Thomas Kuchenbuch. *Filmanalyse Theorien. Methoden. Kritik*. Böhlau Verlag Gesellschaft m. b. H. und Co. KG, ISBN: 3-8252-2648-4, zweite Auflage edition, 2005. 43
- [Lew12] J. R. Lewis. *Handbook of Human Factors and Ergonomics*. G. Salvendy - ISBN: 978-1-118-12908-1, 2012. 149
- [Lik32] R. Likert. *A Technique for the Measurement of Attitudes*. Number Nr. 136-165 in *A Technique for the Measurement of Attitudes*. Science Press, 1932. LCCN 33012634. URL <http://books.google.de/books?id=9rotAAAAAYAAJ>. 147
- [Lop08] Adam Lopez. Statistical machine translation. *ACM Comput. Surv.*, 40(3):8:1–8:49, August 2008. ISSN 0360-0300. URL <http://doi.acm.org/10.1145/1380584.1380586>. 21
- [Lut11] Marco Luthard. *Constraintbasierte 3D Kamerapositionierung*. Diplomarbeit der Hochschule Weihenstephan-Triesdorf, 2011. 12
- [MFK10] Cynthia Matuszek, Dieter Fox, and Karl Koscher. Following directions using statistical machine translation. In *Proceedings of the 5th ACM/IEEE International*

- Conference on Human-robot Interaction, HRI '10*, pages 251–258. IEEE Press, Piscataway, NJ, USA, 2010. ISBN 978-1-4244-4893-7. URL <http://dl.acm.org/citation.cfm?id=1734454.1734552>. 20, 21
- [MHZF12] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. Learning to parse natural language commands to a robot control system. *Proc. of the 13th International Symposium on Experimental Robotics (ISER)*, 2012. 20, 21
- [MKM⁺94] Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert Macintyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*, pages 114–119, 1994. 22
- [MMS93] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2): 313–330, June 1993. ISSN 0891-2017. URL <http://dl.acm.org/citation.cfm?id=972470.972475>. 22
- [MR03] Jacques Marescaux and Francesco Rubino. The ZEUS robotic system: experimental and clinical applications. *Surgical Clinics of North America*, pages 1305–1315, 2003. 16
- [Mül10] Stefan Müller. *Grammatiktheorie*. Number 20 in Stauffenburg Einführungen. Stauffenburg Verlag, Tübingen, 1 edition, 2010. URL <https://hpsg.fu-berlin.de/~stefan/Pub/grammatiktheorie.html>. 24, 29
- [Nou00] Jan Nouza. A large czech vocabulary recognition system for real-time applications. In *Proceedings of the Third International Workshop on Text, Speech and Dialogue, TDS '00*, pages 217–222. Springer-Verlag, London, UK, UK, 2000. ISBN 3-540-41042-2. URL <http://dl.acm.org/citation.cfm?id=647238.720919>. 19
- [OSK10] T. Osa, C. Staub, and A. Knoll. Framework of automatic robot surgery system using visual servoing. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1837–1842, 2010. ISSN 2153-0858. 17
- [Pic02] Jonathan Pickering. *Intelligent Camera Planning in Virtual Environments*. PhD thesis, University of York, 2002. 11

LITERATURVERZEICHNIS

- [Prü93] Jochen Prümper. Software-evaluation based upon iso 9241 part 10. In Thomas Grechenig and Manfred Tscheligi, editors, *Human Computer Interaction*, volume 733 of *Lecture Notes in Computer Science*, pages 255–265. Springer Berlin Heidelberg, 1993. ISBN 978-3-540-57312-8. URL http://dx.doi.org/10.1007/3-540-57312-7_74. 147
- [SB09] J. Stückler and S. Behnke. Integrating indoor mobility, object manipulation, and intuitive interaction for domestic service tasks. *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 506–513, 2009. 22
- [SB11] Mark Steedman and Jason Baldridge. Combinatory Categorical Grammar. In Robert D. Borsley and Kersti Börfjars, editors, *Non-Transformational Syntax - Formal and Explicit Models of Grammar*. Wiley-Blackwell, 2011. Draft 5.0, 2007. 24, 25, 27
- [SEB09] Markus Schächter, Carl E. Eberle, and Nikolaus Brender. *ZDF Jahrbuch. Geschäftsbericht - Jahresrückblick: 2009. Berichtsband (Jahresrückblick 2009)*. ZDF Zweites Deutsches Fernsehen, 1. auflage edition, 2009. ISBN 978-3930610204. 58
- [SL12] Jeff Sauro and R. Lewis, James. *Quantifying the user experience*. Morgan Kaufmann, 1st edition edition, 2012. ISBN 978-0-12-384968-7. 160
- [SN13] Maj Stenmark and Pierre Nugues. Natural language programming of industrial robots. *Robotics (ISR), 2013 44th International Symposium on*, pages 1–5, 2013. 22
- [SP03] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 134–141. Association for Computational Linguistics, Stroudsburg, PA, USA, 2003. URL <http://dx.doi.org/10.3115/1073445.1073473>. 21
- [Ste05] Mark Steedman. Combinatory categorial grammar - introduction. *Linguistics 484 Notes*, 2005. 24, 29
- [TBN00] Bill Tomlinson, Bruce Blumberg, and Delphine Nain. Expressive autonomous cinematography for interactive virtual environments. In *AGENTS '00: Proceedings of*

- the fourth international conference on Autonomous agents*, pages 317–324. ACM, New York, NY, USA, 2000. ISBN 1-58113-230-1. 12
- [THYK03] Takao Tsudo, Yuri Hayakeyama, Shinichi Yoshimura, and Daiichiro Kato. Automatic program production using network-connected robot cameras. *Proceedings of the 2003 IEEE, Las Vegas*, 2003. 13
- [TKD⁺11] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011. 21, 22
- [Tuk77] John W. Tukey. *Exploratory data analysis*. Addison-Wesley, 1. Auflage edition, 1977. ISBN 0-201-07616-0. 157
- [Tur50] Alan M. Turing. Computing Machinery and Intelligence. *Mind*, LIX:433–460, 1950. 18, 37
- [Wag14] Michael Wagner. *Das 1x1 des Videofilms: Grundlagen der Videotechnik*. Fachverlag Schiele & Schön, 2014. ISBN 978-3-7949-0795-3. 19
- [Whi06] Michael White. Ccg chart realization from disjunctive inputs. In *Proceedings of the Fourth International Natural Language Generation Conference, INLG '06*, pages 12–19. Association for Computational Linguistics, Stroudsburg, PA, USA, 2006. ISBN 1-932432-72-8. URL <http://dl.acm.org/citation.cfm?id=1706269.1706274>. 30
- [WLY11] Boaccheng Wang, Zhijun Li, and Weijun Ye. Speech control of a teleoperated mobile humanoid robot. *International Journal of the physical sciences*, 6 (18), 2011. 22
- [WM06] Y. W. Wong and R. J. Mooney. Learning for semantic parsing with statistical machine translation. *Proc. of the main conference on Humand language Technology Conf of the North American Chapter of the Association of Computational Linguistics*, 2006. 20
- [Yeh53] Bar-Hillel Yehoshua. A quasi-arithmetical notation for syntactic description. *Language*, 29:47–58, 1953. 24