

Intelligente Systeme und Robotik

Computer-controlled human soccer

Björn Karger

Matr. 4199740

Betreuer: Prof. Dr. Raúl Rojas
Betreuender Assistent: Dr. Hamid Mobalegh

Institut für Informatik, Freie Universität Berlin, Deutschland

18. November 2013

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, sind als solche gekennzeichnet. Die Zeichnungen oder Abbildungen sind von mir selbst erstellt worden oder mit entsprechenden Quellennachweisen versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner Prüfungsbehörde eingereicht worden.

Berlin, den 18. November 2013

(Björn Karger)

Zusammenfassung

Zusammenfassung

Die vorliegende Masterarbeit befasst sich mit der Konstruktion einer Hardwareplattform, welche einen Menschen mit zusätzlichen Sensoren ausstattet und eine autonome Verarbeitung erlaubt. Zweck der Konstruktion soll es sein, einen Menschen als alternative, humanoide Roboterplattform für Fußballspiele zu nutzen, welche im Rahmen des RoboCups stattfinden. Die Verwendung eines Menschen bezweckt eine Loslösung von wartungsintensiven Roboterplattformen, bei denen eine Interaktion mit der Umwelt an viele Bedingungen geknüpft ist. Eine optimale Funktion der Mechanik, Elektronik und Software wird benötigt, um eine gewünschte Interaktion zu erbringen, welche ein Mensch ohne Voraussetzungen vollbringt. Die gewonnene Zeit, welche durch die Abhängigkeitsreduktion entsteht, soll für die Weiterentwicklung der verbleibenden Themenfelder oder zur Einführung neuer Problemstellungen verwendet werden, um so eine beschleunigte Entwicklung zu erreichen. Die Plattform Mensch soll eine Roboter-Plattform simulieren, wodurch es den Einfluss des Menschen zu minimieren gilt. Dafür werden ihm die Augen verbunden und Befehle per Audio-Signal übermittelt, welche es umzusetzen gilt.

Ziel dieser Arbeit ist die Konstruktion einer solchen Plattform, die Anpassung und Neuentwicklung benötigter Software sowie die Demonstration der Ergebnisse auf dem RoboCup 2013. Als Basis der Hardware sowie Software dienen die humanoiden Roboter der FUManoids, welche für das neue Szenario angepasst wurden.

Abstract

In this master thesis a new hardware platform was developed, which has the goal to replace the movement hardware of a humanoid robot with a human. Due to many dependencies it is necessary to make sure that hardware, electronics and software are working together perfectly to achieve certain goals, like to make a robot walk. To reduce the amount of time spent on maintenance, a human should provide the necessary movement skills. To reduce the influence of the human, he/she will be blindfolded and will receive movement commands by audio signals. The saved time can be spent to accelerate the development of soccer playing robots and maybe introduce new challenges. The platform has the same sensory input as common humanoid robots and was presented at the RoboCup 2013.

Inhaltsverzeichnis

1	Einleitung	1
1.1	RoboCup	2
1.2	Motivation und Ziele	4
1.3	Team FUmanoids	6
2	Grundlagen	7
2.1	Wahrnehmung	7
2.1.1	Tastsinn	8
2.1.2	Hörsinn	9
2.2	Berlin United-Framework	13
2.3	Perzepte und Modelle	14
2.4	Roll-Pitch-Yaw-Winkel	16
2.5	XABSL	16
2.6	OpenAL	18
3	Ähnliche Arbeiten	21
3.1	3D-Audio im Flugverkehr	21
3.2	3D-Audio-Schnittstelle für Blinde	22
3.3	3D-Audio für erweiterte Realität	23
3.4	Oculus Rift	24
4	Hardware	25
4.1	Prototyp 1 und 2	25
4.2	RoboCup-Prototyp	27
4.2.1	Kamera	28
4.2.2	Minicomputer	28
4.2.3	Akkumulator	29
4.2.4	Inertialsensor (IMU)	30
4.2.5	Pegelwandler	31
4.2.6	Arbeitsschutzbrille	33
4.2.7	Gehörschutzkopfhörer	34
4.3	Aufbau	35
4.4	Kosten	36

5	Software	37
5.1	Entwicklungsumgebung	37
5.2	WLAN-Verbindung	38
5.3	Audio-Framework	38
5.4	IMU - UM6	40
5.5	Vision	40
5.6	Modellierung	41
5.7	Direct Mode	41
	5.7.1 Blindenfussball	41
	5.7.2 Umsetzung	42
5.8	Command Mode	44
	5.8.1 CCHS-Symbole	44
	5.8.2 Verhalten auf dem RoboCup 2013	47
	5.8.3 Demonstration auf dem RoboCup 2013	47
	5.8.4 Regeln	50
6	Zusammenfassung und Ausblick	55
6.1	Zusammenfassung	55
6.2	Ausblick	56
	6.2.1 Hardware	56
	6.2.2 Software	57
	6.2.3 RoboCup	59

KAPITEL 1

Einleitung

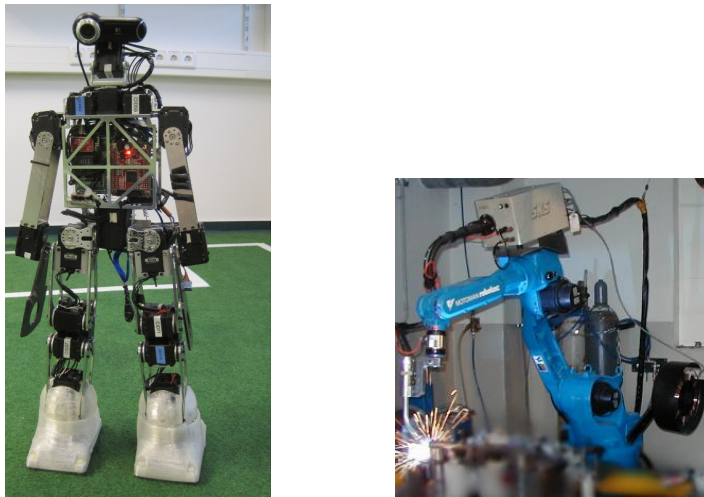


Abbildung 1.1: Ein humanoider Roboter [1] (links) und ein Industrieroboter ¹(rechts)

Heutzutage ist es nicht mehr ungewöhnlich, wenn man bereits Kontakt mit Robotern hatte. Sei es als Spielzeug für interessierte Personen, als Arbeitsgerät in einer Industrieanlage, als Spielgefährte aus futuristischen Filmen oder aber als Forschungsobjekt einer Universität, der Kontakt mit Robotern ist allgegenwärtig. Die Spanne der Roboter ist jedoch genauso weit gefächert wie die Definition des Begriffs „Roboter“ selbst, welche sich im Wandel der Zeit entsprechend geändert hat. So ist der Begriff in einem Lexikon von 1975 unter anderem als „i. allg. Bezeichnung für elektronisch gesteuerte Geräte“ [2] definiert, wonach moderne Waschmaschinen als Roboter einzuordnen sind. Diese Einordnung stimmt trotz geänderter Definition auch nach heutigen Maßstäben, was die Begriffserklärung aus dem Duden „Automat, der fernge-

¹<http://commons.wikimedia.org/wiki/File:Industrieroboter.jpg>

steuert oder nach Sensorsignalen bzw. einprogrammierten Befehlsfolgen anstelle eines Menschen bestimmte mechanische Tätigkeiten verrichtet“² zeigt. Um die Entwicklung der Künstlichen Intelligenz, wozu auch die Robotik zählt, weiter voranzutreiben, hat sich die Wissenschaft als Ziel gesetzt, ein humanoides Fußballteam aufzustellen und gegen Menschen antreten zu lassen, was die Kombination und Zusammenarbeit von mehreren Fachgebieten erfordert. Dieses gesetzte Ziel wird im Rahmen des RoboCups stufenweise umgesetzt, welcher im nächsten Abschnitt näher erläutert wird.

1.1 RoboCup

Seit dem Jahr 1997 wird jährlich die RoboCup-Weltmeisterschaft³ abgehalten. Das langfristige Ziel dabei ist es, mit autonomen, humanoiden Robotern die zukünftigen Weltmeister bis Mitte des 21. Jahrhunderts nach offiziellen FIFA-Regeln zu besiegen. Um dieses Ziel zu erreichen werden jährlich die Regeln der einzelnen Ligen angepasst und die Anforderungen an die Roboter erhöht. Im Laufe der Zeit haben sich weitere Forschungsziele etabliert, wodurch das originale Ziel nun „RoboCup Soccer“ genannt wird. Neben diesem gibt es „RoboCup@Home“, bei dem Assistenzroboter für den Heimeinsatz entwickelt werden, „RoboCup Rescue“, wo Roboter in Katastrophenszenarien behilflich sein sollen und „RoboCupJunior“, der Schüler an das Thema Robotik heranführen soll. Jedes dieser Ziele gliedert sich in einzelne Ligen auf, welche durch individuelle Regeln unterschiedliche Aspekte dieser Teilgebiete erforschen. Diese Arbeit findet sich im Rahmen des „RoboCup Soccer“ wieder, wodurch im Folgenden die fünf zugehörigen Ligen erläutert werden.

Small Size In der Small-Size-Liga treten kleine Roboter mit maximal 18 Zentimetern Durchmesser und 15 Zentimetern Höhe gegeneinander an. Die Roboter erhalten Bewegungsbefehle von außerhalb des Feldes, wo ausgelagerte Computer die Berechnungen übernehmen und entsprechende Kommandos per Funksignal übermitteln. Die Spielsituation wird über hängende Kameras erfasst und allen Spielparteien in einem standardisiertem Format für weitere Verarbeitung zur Verfügung gestellt, sodass eine eigene Erkennung der Spielobjekte entfällt. Der Fokus dieser Liga liegt auf der Konstruktion von kompakten Robotern und auf dem Zusammenspiel der einzelnen Roboter in einer sich schnell verändernden Umgebung.

Middle Size Im Gegensatz zur Small-Size-Liga transportieren die wesentlich größeren Roboter ihre Sensoren selbst und werten die erfassten Daten selbstständig aus. Dennoch steht für weitere Berechnungen und die Koordination der Roboter ebenfalls ein zentraler Computer bereit. Der Middle-Size-Liga steht aufgrund der großen Ausmaße von Roboter sowie Ball und einer Spieleranzahl von fünf Robotern pro Team das größte Spielfeld mit 18 mal 12 Metern zur Verfügung.

²<http://www.duden.de/rechtsschreibung/Roboter>

³<http://www.robocup.org/>

Simulation In der Simulationsliga wird mit virtuellen Robotern (Agenten) auf einem 2D- oder 3D-Spielfeld gespielt, welche von einem zentralen Computer bestimmte Informationen über ihre eigene Umwelt erhalten und diesem im Anschluss eine umzusetzende Aktion mitteilen. Die Informationen, die der Agent erhält, sind allerdings keine Idealwerte, sondern sind mit Rauschen versetzt und nur aus der Perspektive des Agenten verfügbar, um eine Annäherung an die Realität zu gewinnen. Da die Konstruktion eines Roboters bei dieser Liga entfällt, kann verstärkt auf die Umsetzung neuer Algorithmen der Künstlichen Intelligenz und der Anwendung einer Team-Strategie geachtet werden.

Standard Platform Bei der Standard-Platform-Liga werden kommerzielle Roboter eines bestimmten Modells für alle Teilnehmer vorgeschrieben, sodass sich ausschließlich auf die Entwicklung der Software konzentriert werden kann. Aktuell wird für diesen Zweck der NAO von Aldebaran Robotics⁴ verwendet, welcher im Bild 1.2 dargestellt ist. Der Roboter agiert auf dem Spielfeld autonom und er ist humanoid, wodurch sie eigene Spielentscheidungen aufgrund der analysierten Sensordaten treffen sowie sich auf zwei Füßen fortbewegen müssen. Die Ressourcen in solch einem Roboter sind begrenzt, wodurch mit der zur Verfügung stehenden Rechenleistung sparsam umgegangen werden muss.



Abbildung 1.2: Standard-Plattform-Roboter NAO⁵

Humanoid Die Humanoid-Liga [3] ist die Steigerung der Standard-Platform-Liga, bei dem die Roboter kein einheitliches Modell besitzen. Die Roboter können nach bestimmten Regeln konstruiert werden oder kommerziell erworben werden. Der Aufwand ist durch den Hardwareanteil wesentlich höher als bei der Standard-Platform-Liga, da die Hardware entworfen, konstruiert und gewartet werden muss. Die Roboter müssen eine menschliche Gestalt nach bestimmten Maßen vorweisen, wie im linken Bild 1.1 ersichtlich. Dadurch sind künstliche sowie physikalische Begrenzungen vorhanden, die sowohl Sensoren, Aktuatoren

⁴<http://www.aldebaran-robotics.com/en/>

⁵<http://www.aldebaran-robotics.com/en/Pressroom/Photography/nao.html#10>

(Motoren), Rechenleistung als auch die Kapazität der Energiequellen limitieren. Diese Liga verlangt nach einer optimalen Zusammenarbeit der Software und Hardware, um das Ziel eines fußballspielenden Roboters zu erreichen.

1.2 Motivation und Ziele



Abbildung 1.3: Demonstration auf dem RoboCup 2013

Bei der Entwicklung von humanoiden Robotern und deren Funktionalitäten ist bei der Gestaltung des Verhaltens, welches auf höheren Softwareebenen entworfen wird, immer eine implizite Abhängigkeit von darunterliegenden Schichten vorhanden. So kann zum Beispiel der Test einer neuen Dribbling-Methode durch eine gelockerte Schraube in einem Gelenk dazu führen, dass der Test wider Erwarten negativ ausfällt und eine lange Fehlersuche nach sich zieht. Es existiert also ein Abhängigkeitsbaum, welcher bei autonomen Robotern von Elektronik über Mechanik bis hin zur Software aus hunderten Komponenten besteht. Für einzelne Komponenten können jeweils unterschiedliche Personen zuständig sein und die optimale Funktion aller Komponenten muss sicher gestellt werden, um das gewünschte Ergebnis zu liefern. Einer der aufwändigsten und zeitintensivsten Schritte ist daher die Sicherstellung der optimalen Funktionalität von allen Teilen dieses Abhängigkeitsbaums, wobei damit wichtige Zeit für die eigentliche Forschung verloren geht.

Daher hat Dr. Hamid Mobalegh mit „Computer controlled human soccer“(CCHS) einen neuen Ansatz vorgestellt, wie ganze Zweige des Abhängigkeitsbaums losgelöst werden können. Die Grundidee dabei ist, statt Robotern, welche viele potenzielle Fehlerquellen haben, einen Menschen als Plattform für eine minimale Hardwareaus-

stattung zu nutzen. So können Software und Hardware nahezu losgelöst von einander weiterentwickelt werden, sodass Hardwarefehlerquellen bei der Softwareentwicklung nahezu ausgeschlossen werden können. Der Mensch ist als Ersatz optimal geeignet, da er das eigentliche Ziel der Roboterentwicklung darstellt und somit alle nötigen Fähigkeiten, wie zum Beispiel einen stabilen Gang, mit sich bringt. Ein weiterer Vorteil ergibt sich durch die reduzierte Hardware am Menschen, welche nur einen Bruchteil eines normalen humanoiden Roboters kostet und zudem leichter zu warten ist. Dadurch könnte die Beteiligung am RoboCup und somit auch an der Forschung steigen, da die Finanzierung eines Teams erleichtert wird. Auch könnten größere Mannschaften antreten, wodurch eine Annäherung an die normalen Fußballverhältnisse von 11 Spielern pro Mannschaft möglich erscheint. Zusätzlich könnte durch eine verstärkte Einbindung des Menschen ein gesteigertes Interesse am RoboCup geschaffen werden, welches der Forschung zugute kommt. Mit der Plattform wird es möglich auch unpräparierte Spielfelder zu nutzen, da der Mensch wesentlich flexibler im Umgang mit seiner Umgebung ist als heutige Roboter. Dadurch ergeben sich unter anderem neue Herausforderungen für die Bildverarbeitung, welche ohne fest definierte Farben alle Objekte erkennen muss oder aber sogar mit ständigen Änderungen der Lichtverhältnisse auf einem Außenspielfeld zurechtkommen muss.

Im Rahmen dieser Masterarbeit soll gezeigt werden, dass dieses Konzept umsetzbar ist, wofür Arbeiten an Hardware sowie Software nötig sind. Als Proof of Concept dienen mehrfache Demonstrationen (siehe Abbildung 1.3) auf dem RoboCup 2013, welche in einem späteren Abschnitt (5.8.3) separat behandelt werden. Die Demonstrationen sollten das Potenzial der Plattform zeigen sowie Öffentlichkeitsarbeit leisten. Ein Antrag auf Fördermittel wurde von der RoboCup Föderation bewilligt, welche zum Kauf der benötigten Hardware dienen. Als Basis für die Hardware wurde die Roboter-Plattform und die Erfahrung des Teams „FHumanoids“ genutzt. Auch bei der Software diente das Programm der „FHumanoids“ als Basis für die Prototypen. In dieser Arbeit wurden mehrere Prototypen entworfen, gebaut und die Software entsprechend angepasst. Die finale RoboCup-Version wurde in entsprechender Stückzahl hergestellt, um die Demonstrationen auf dem RoboCup 2013 zu veranstalten.

Was macht ein Mensch, wenn seine vielleicht wichtigste Wahrnehmung, die visuelle Wahrnehmung, blockiert wird? Wie kann der fehlende Sinn adäquat kompensiert werden? Wie reagiert ein Mensch auf die Fernsteuerung durch einen Computer? Diese und ähnliche Fragen mussten bei der Umsetzung dieser Arbeit beachtet werden. Die folgenden zwei, diametralen Ansätze wurden verwirklicht, um die Fähigkeiten der Plattform zu demonstrieren.

Direct Mode Der Direct Mode soll die Möglichkeiten der Plattform nutzen, um Menschen mit Sehbehinderung das Fußballspielen zu ermöglichen. Ziel hierbei ist, dem Spieler eine möglichst hilfreiche Repräsentation der Wirklichkeit durch verschiedenste Methoden zu vermitteln, auf denen eigene Entscheidungen aufbauen. Hier wird untersucht, welche erkannten Objekte in welcher Form dem Nutzer vermittelt werden müssen, um diesem eine Orientierung auf dem Spiel-

feld zu ermöglichen. Zum Beispiel kann entschieden werden, ob das eigene Tor mit einem Piepton, einer Melodie oder aber anderweitig darzustellen ist und ob die Darstellung des eigenen Tors überhaupt vonnöten ist. Solche Erkenntnisse können ebenfalls zur Verbesserung des folgenden „Command Mode“ dienen.

Command Mode Im Command Mode geht es um die Nutzung des Menschen als Roboterplattform, wobei der Einfluss des Menschen minimiert werden soll. Der Mensch erhält in diesem Modus ausschließlich durch den Computer gefilterte Informationen, welche keinen Rückschluss auf die Wirklichkeit zulassen sollten, da sonst der menschliche Faktor nicht kontrollierbar ist. Der Mensch bekommt also nur Befehle was er zu tun hat und soll diesen Befehlen Folge leisten, um die normale Funktion eines Roboterkörpers nachzuahmen. Um den Menschen möglichst wenig bei der Entwicklung der Robotersoftware beachten zu müssen, sollte eine einfache Schnittstelle dafür sorgen, dass der Mensch wie gewöhnliche Roboterhardware zu steuern ist. Trotz der gegenteiligen Ziele des Command Mode und des Direct Mode können einige Erkenntnisse übertragen werden, was beim Command Mode zum Beispiel zu einer verbesserten Reaktionszeit des Menschen und somit zu einem verzögerungsärmeren System, also einer Annäherung an elektromechanische Hardware, führen könnte.

1.3 Team Fumanoids



Abbildung 1.4: Logo des Team Fumanoids⁶

Das Fumanoids Team⁷ ist ein wissenschaftliches Projekt der Freien Universität Berlin unter akademischer Leitung, welches im Jahr 2006 gegründet wurde und seit dem Folgejahr am RoboCup Soccer teilnimmt. Die Fumanoids treten in der humanoiden Kid-Size-Liga an, wo sie bereits mehrere Erfolge verbuchen konnten und werden hauptsächlich in Form von Abschlussarbeiten weiterentwickelt. Im linken Bild 1.1 ist der humanoide Roboter aus dem Jahr 2012 dargestellt, auf welchem diese Arbeit aufbaut.

⁶<http://www.fumanoids.de/about/>

⁷<http://www.fumanoids.de/>

KAPITEL 2

Grundlagen

2.1 Wahrnehmung

Ein Mensch erfasst alle seine Umwelteinflüsse durch spezialisierte Rezeptoren. Dafür stehen mehr als 100 Millionen Rezeptoren zur Verfügung, welche dem 100- bis 1000-fachen an Neuronen im Gehirn gegenüberstehen. Die Neuronen werten die Impulse der Rezeptoren aus und interpretieren diese, woraufhin entsprechende Aktionen ausgelöst werden. Diese Datenmenge muss reduziert werden, um adäquat verarbeitet werden zu können, wodurch es zu manchen Fehlinterpretationen der Daten kommen kann. Die bekanntesten Wahrnehmungstäuschungen sind, aufgrund ihrer hohen Anzahl, die optischen Illusionen (siehe Bild 2.1). Die restlichen Sinne lassen sich ebenso gezielt täuschen, wofür teils aufwendigere Methoden nötig sind. Beim Tastsinn kann es zu dem Eindruck kommen, dass identische Massen mit unterschiedlichen Volumen auch unterschiedliche Massen besitzen, da in der Regel das Objekt mit der höheren Dichte schwerer erscheint. Daher muss die Funktionsweise der Sinne und deren Beschränkungen bei der Interaktion des Computers beachtet werden, um eine möglichst eindeutige Kommunikation zu ermöglichen. [4]

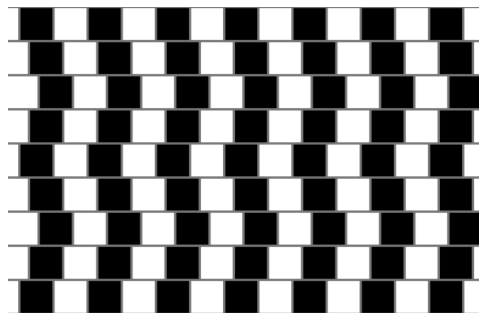


Abbildung 2.1: Die horizontalen Linien erscheinen gebogen, sind jedoch parallel. ¹

Die visuelle Wahrnehmung, welche elektromagnetische Wellen eines gewissen Spektrums durch die Augen erfasst, wird bei einem normalem Fußballspiel ausgiebig zur Identifizierung wichtiger Objekte sowie zur Lokalisierung auf dem Spielfeld genutzt. Durch das vorgestellte Konzept ist dieser Sinn jedoch nicht mehr Verfügbar und muss daher nicht näher betrachtet werden.

Ein weiterer wichtiger Sinn ist die auditive Wahrnehmung durch die Ohren. Dabei werden Schallwellen (Luftdruckänderungen) in Signale umgewandelt, um vom Gehirn interpretiert zu werden. Dieser Sinn stellt einen Kernpunkt der menschlichen Kommunikation dar und ist ein wichtiger Teil dieser Arbeit. Er ermöglicht eine Kommunikation mit einem Computer und ist für die Kompensation des fehlenden Sehsinns geeignet, was sehbehinderte Menschen im alltäglichen Leben beweisen. Der Hörsinn wird im Abschnitt 2.1.2 näher betrachtet.

In der klassischen Wahrnehmungslehre verbleiben der Tastsinn, der Geruchssinn und der Geschmackssinn, wobei die Letzteren kaum einen Einfluss auf ein Fußballspiel haben sollten. Der Tastsinn hingegen ist für die Handhabung des Balls unerlässlich, wodurch eine nähere Betrachtung sinnvoll erscheint. In der modernen Wahrnehmungslehre existieren noch weitere Sinne, wie der Gleichgewichtssinn oder die Körperempfindung. Der Gleichgewichtssinn wird für eine stabile Fortbewegung benötigt, kann aber nicht gezielt beeinflusst werden, wodurch von einer Betrachtung abgesehen wird. [4]

2.1.1 Tastsinn

Der Tastsinn unterteilt sich in Sensorengruppen für Schmerz, Temperatur und Bewegung. Jede dieser Gruppen besitzt spezialisierte Rezeptoren, um fest definierte Reize aufzunehmen und elektrisch an das Gehirn weiter zu leiten. Die Zuordnung der unterschiedlichen Reize erfolgt dabei durch die Anbindungsstelle im Gehirn. Für die einzelnen Informationsleitungen ist also bekannt, welche Informationen in ihnen enthalten sind, wodurch es zum Beispiel zu keinen Verwechslungen von Berührungs- und Temperaturempfindungen kommt. Durch die Vielfalt und Verteilung der Rezeptoren ergibt sich ein möglicher Kommunikationskanal zwischen Mensch und Computer, welcher bereits durch den Vibrationsalarm bei Mobiltelefonen genutzt wird. [4]

Bewegung

Die Sensorengruppe für Bewegung umfasst Sensoren für alle mechanischen Reize wie Druck, Dehnung, Vibration. Für die Detektion dieser Reize sind die Merkel-Zellen, Meissner-, Ruffini- und Vater-Pacini-Körperchen zuständig, wobei ihre Funktion durch ihren Aufbau bestimmt wird. Die Merkel-Zellen reagieren auf Druck, adaptieren jedoch nur langsam auf den dargebrachten Reiz, wobei die Signalfrequenz proportional zur Reizintensität steigt und somit eine Druckintensität vermittelt wird.

¹http://commons.wikimedia.org/wiki/File:Café_wall.svg, CC BY-SA 3.0

Die Meissner-Körperchen sind hingegen schnell adaptierende Drucksensoren, welche hauptsächlich auf Druckänderungen reagieren. Die Ruffini-Körperchen sind langsam adaptierende Dehnungssensoren und helfen beim Feststellen der Gelenkstellungen. Die Vater-Pacini-Körperchen adaptieren sehr schnell und reagieren auf Vibrationen der Haut. [4]

Temperatur

Die Temperatursensoren sind unterteilt nach Kälte- und Wärmerezeptoren, welche bei entsprechender Reizung verstärkt Signale an das Gehirn senden. Durch diese Sensoren ist es dem Menschen möglich auf eine veränderte Umgebungstemperatur zu reagieren und sich entsprechend zu kleiden. Dadurch kann der Mensch seine Körpertemperatur einfacher bei 37 Grad halten als durch Zittern oder Schwitzen, was die Aufwendung von Energie zur Folge hätte. In gewissen Temperaturbereichen erfolgt eine Gewöhnung an die sensorischen Werte, wodurch die Temperatur wieder als normal wahrgenommen wird, sodass kein ständiger Wechsel zwischen den Temperaturempfindungen erfolgt, obwohl die Sensoren bis auf Zehntel Grad genau messen. [4]

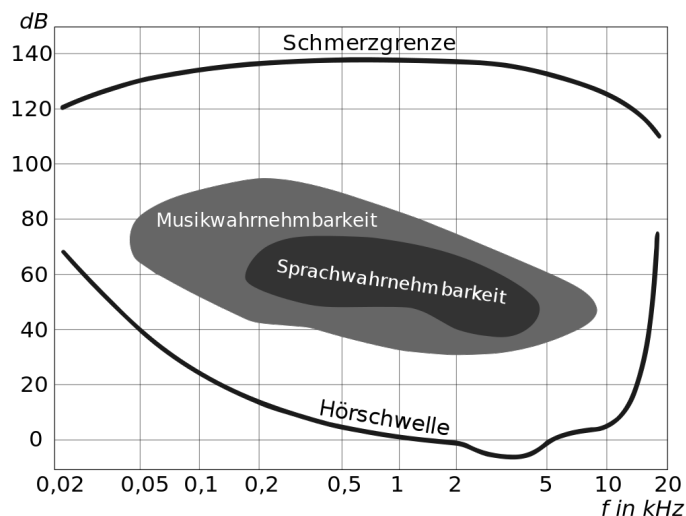
Schmerz

Die Schmerzsensoren dienen zum Signalisieren von Verletzungen, welche in der weiteren Verarbeitung durch entsprechende Reaktionen reduziert werden können. So kann eine schnelle Reaktion auf einen Schmerz beim Umgang mit einer Gefahrenquelle schwerere Verletzungen vermeiden. Der Schmerzsensor (Nozizeptor) erzeugt mit unterschiedlicher Schnelligkeit Signale, welche im Gehirn weit gefächert verarbeitet werden. Ein schneller Schmerzreiz wird zum Beispiel beim Stoßen des kleinen Zehs ausgelöst, welcher aber relativ schnell wieder nachlässt. [4]

2.1.2 Hörsinn

Die Ohren nehmen Luftdruckunterschiede (Schall) durch circa 15000 Haarzellen pro Ohr wahr, welche durch Änderungen des Luftdrucks erregt werden und über Nervenfasern entsprechende Signale zum Gehirn weiterleiten. Die Luftdruckänderungen werden sowohl durch ihre Frequenz sowie durch ihren Schalldruck repräsentiert, welche durch die Haarzellen erfasst werden. Beide Werte sind vergleichbar mit Wellen auf einem See, wobei der Abstand der Wellenberge die Periodendauer (Kehrwert der Frequenz) darstellt und der Schalldruck die Höhe der Wellenberge. Das Gehirn ist nun in der Lage durch die Beobachtung von zwei Stellen am Ufer mehrere wellenverursachende Schiffe zu orten und in ihrer Größe einzuordnen. [4]

Hörfläche

Abbildung 2.2: Hörfläche eines normalhörenden Menschen ²

Im Bild 2.2 ist die Hörfläche eines normalhörenden Menschen dargestellt, welche den wahrnehmbaren Schall in Bezug auf Schalldruckpegel (Intensität) und Frequenz (Tonhöhe) definiert. Die jeweiligen Ober- und Untergrenzen unterscheiden sich dabei von Mensch zu Mensch und werden zum Beispiel durch das Alter oder Krankheiten beeinflusst. Die für einen Menschen hörbaren Frequenzen befinden sich zwischen 16 Hertz (Hz) und 20 Kilohertz (kHz). Frequenzen unter 16 Hz werden Infraschall genannt und sind für manche Tiere wie Elefanten wahrnehmbar. Frequenzen über 20 kHz werden Ultraschall genannt und dienen zum Beispiel Fledermäusen zur Orientierung. Die Hörschwelle definiert gerade noch wahrnehmbare Töne und ist abhängig von der Frequenz. Man erkennt in den Frequenzgrenzbereichen, dass eine wesentlich höhere Intensität nötig ist, um einen Ton noch wahrnehmen zu können. Nach oben hin ist die Hörfläche durch die Schmerzgrenze beschränkt, oberhalb derer es zu bleibenden Hörschäden kommen kann. Vergleicht man die Musik- und Sprachwahrnehmung, welche im Wesentlichen für die jeweiligen Zwecke genutzt werden, dann lässt sich feststellen, dass die Sprache des Menschen nur eine Teilmenge der Musikwahrnehmung ist. Die Nutzung von Tönen außerhalb der Sprachwahrnehmung erhöht somit die Bandbreite bei der Informationsübermittlung und sollte beim Entwurf von effizienter Kommunikation nicht außer acht gelassen werden. Auch darf das individuelle Hörvermögen der Menschen, welche die Befehle des Computers umsetzen sollen, nicht unbeachtet bleiben, falls Töne in den Grenzbereichen der Hörfläche verwendet werden. [5]

Ab einem Pegel von 10 dB über der Hörschwelle wird ein Ton als Signal wahrgenommen.

²<http://commons.wikimedia.org/wiki/File:Hoerflaeche.svg>, gemeinfrei

nommen und ab 15 dB ist ein Signal kaum noch zu überhören. Daher sollte sich ein Signal in einem Pegelbereich von 15 dB bis 25 dB über der Hörschwelle befinden, um eine optimale Wahrnehmung zu erreichen. Eine weitere Erhöhung der Intensität führt nicht zu einer verbesserten Wahrnehmung und kann die Wahrnehmung weiterer Signale stören. [6]

Ortung

Die Ortung von Tönen wird unterschieden nach Richtung und Entfernung. Bei der Entfernungsortung kann vereinfacht gesagt werden, dass die vermutete Entfernung hauptsächlich vom Schalldruckpegel des Tons abhängt, egal wie weit die Tonquelle wirklich vom Hörer entfernt war. Laute Objekte in größerer Entfernung können den gleichen Entfernungseindruck vermitteln wie leisere Objekte, welche sich näher am Hörer befinden. [5]

Bei der Richtungsortung hingegen spielen sehr viele Faktoren, wie Frequenz, Ort, Anatomie des Menschen und vieles mehr eine Rolle. Für die Behandlung der Richtungsortung ist zunächst eine Definition und Benennung der benötigten Achsen nötig, was im folgenden Bild 2.3 dargestellt wird. Je nach Fachbereich werden diese Achsen unterschiedlich bezeichnet, was besonders im folgenden Abschnitt 2.4 deutlich wird. [5]

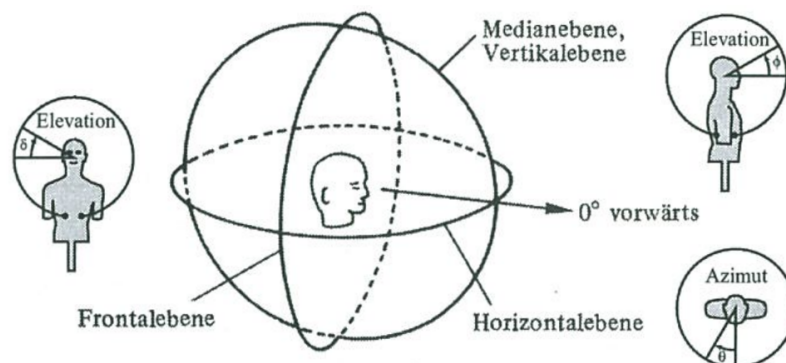


Abbildung 2.3: Achsen des Kopfes und ihre Benennung [5]

Die meisten Faktoren zur Richtungsortung sind bezogen auf das Hören mit beiden Ohren, wobei auch eine Richtungsortung mit nur einem Ohr erfolgen kann. Mit nur einem Ohr spielt das Frequenzspektrum der Tonquelle und die Vertrautheit mit dem Ton eine entscheidende Rolle, wodurch es möglich wird, grob die Richtungen zu bestimmen, allerdings mit einem hohen Fehler. Eine wesentlich genauere Richtungsortung ist mit beiden Ohren möglich, da hier zusätzliche Informationen wie Laufzeitunterschiede zwischen den Ohren ausgewertet werden können. Die Laufzeitunterschiede entstehen, sobald sich die Tonquelle nicht auf der Medianachse befindet, also mit einem seitlich gedrehtem Kopf zur Quelle. Durch die unterschiedlichen Entfer-

nungen der einzelnen Ohren zur Quelle entsteht eine Phasenverschiebung des Signals. Basierend auf den Laufzeitunterschieden kann die Position der Quelle jedoch nicht eindeutig bestimmt werden, was der sogenannte „Cone of Confusion“ verdeutlicht. Wie im Bild 2.4 zu erkennen, gibt es einen Kegel (Cone) auf dessen Oberfläche unterschiedliche Tonquellen identische Laufzeitdifferenzen besitzen. Hierdurch kommt es zu Verwechslungen, sodass zum Beispiel eine Tonquelle, die sich hinter einer Person befindet, von vorne zu kommen scheint. [5]

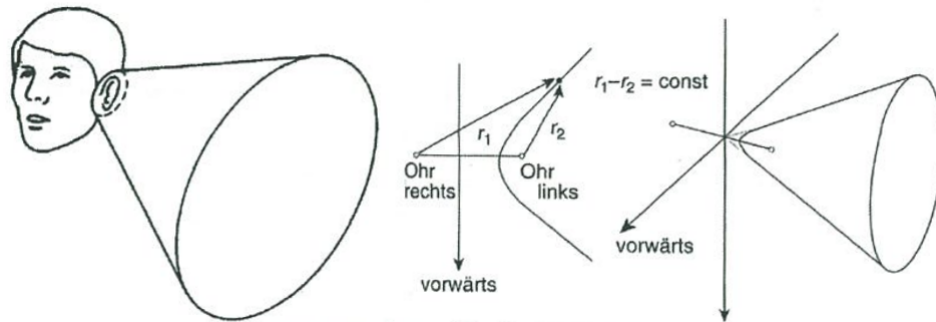


Abbildung 2.4: Darstellung eines Cone of Confusion. [5]

Betrachtet man die Pegeldifferenzen, also die Intensitätsunterschiede, ergibt sich die gleiche Mehrdeutigkeit, wobei die Intensität zusätzlich von der Frequenz des Signals abhängig ist. Es entsteht mit zunehmender Frequenz ein stärker werdender, akustischer Schatten auf der abgewandten Seite zur Tonquelle, der durch das Verhältnis von Wellenlänge zur Kopfgröße bestimmt wird und durch Absorption entsteht. Um solche Mehrdeutigkeiten aufzulösen, werden andere Faktoren wie zum Beispiel die Form des Außenohrs oder sogar die Schulter genutzt, da diese eine richtungsabhängige Reflektoreigenschaft aufweisen und zur besseren Bestimmung des Azimut- und Elevationswinkels beitragen. Trotz all dieser Korrekturen kann es zur Fehlortung einer Tonquelle kommen, was nicht verwunderlich ist, da mit zwei eindimensionalen Informationsquellen die Rekonstruktion einer dreidimensionalen Welt erfolgt. [5]

Die Richtungsortung von Objekten verbessert sich jedoch, wenn der Kopf bewegt wird und die Kopfposition durch das Gleichgewichtsorgan, Muskelanspannungen und optisches Feedback genau bestimmt werden kann. Durch die zusätzliche Information über die Änderungswinkel des Kopfes lassen sich viele Mehrdeutigkeiten beseitigen. Grundsätzlich kann gesagt werden, dass Objekte, die sich vor dem Menschen befinden, genauer geortet werden können und dass die Richtungsortung auf der Horizontalebene genauer erfolgt als auf den restlichen Ebenen. [5]

2.2 Berlin United-Framework

Das Berlin United-Framework [7] ist eine Software-Plattform, welches ein Grundgerüst für unterschiedlichste Projekte bereitstellt. Das Framework wird von mehreren Teams, wie den FUmoids, dem Nao Team Humboldt³ und dem Racing Team⁴ der FU-Berlin genutzt. Das Framework basiert auf einem Blackboard-System, welches komplexe Probleme in Teilprobleme aufteilt, die wiederum durch spezifische Lösungen bearbeitet werden. Das Blackboard ist in diesem Fall die Sammlung vieler Repräsentationen, welche unterschiedlichste Daten, wie Spielfeldkontur, IMU-Daten oder Motorwerte speichern. Diese Repräsentationen sind die Zwischenspeicher für Module, welche die eigentlichen Berechnungen durchführen. Module melden das Lesen einer Repräsentation mit **REQUIRE** an und erhalten durch **PROVIDE** Lese- sowie Schreibrechte. Durch diese Definition von Modul- und Repräsentationsabhängigkeiten entsteht ein gerichteter, azyklischer Graph (siehe Abbildung 2.5), welcher durch den Modulmanager aufgebaut und entsprechend abgearbeitet wird. Ein Modul, welches nur Daten in eine Repräsentation speichert, ist ein Startpunkt eines solchen Graphen (orange), woraufhin andere Module (blau) die Daten dieser Repräsentation weiterverarbeiten und wiederum in neue Repräsentationen (beige) speichern. Nachfolgend ist die Moduldefinition des FieldContourExtractor dargestellt (Quellcode 2.1) und im Bild 2.5 der entsprechende Graph mit allen Abhängigkeiten dargestellt.

```
1 BEGIN_DECLARE_MODULE(FieldContourExtractor)
2     REQUIRE(Image)
3     REQUIRE(CalibrationFile)
4     REQUIRE(ImageDimensions)
5     REQUIRE(FieldColor)
6     REQUIRE(CameraMatrix)
7
8     PROVIDE(FieldContour)
9 END_DECLARE_MODULE(FieldContourExtractor)
```

Quellcode 2.1: Moduldeklaration

Das Framework ist in zwei Komponenten aufgeteilt, einen allgemeinen Teil und einen projektspezifischen Teil. Der Erstere stellt Basisfunktionen wie Kommunikation, Modulmanager, Debugging-Optionen und allgemeine Hardwareansteuerung, wie Kamera- oder Motoransteuerung, bereit, welche projektübergreifend genutzt werden können. Im Letzteren werden, entsprechend dem angestrebten Projekt, die Module und Repräsentation angelegt und implementiert. Für ein Projekt, wie die FUmoids, ergibt sich durch das Framework der Vorteil, dass neue Teammitglieder, welche einen abgegrenztes Problem behandeln, nur mit den notwendigen Repräsentationen arbeiten, die für die Erfüllung des Teilproblems benötigt werden. Die Einarbeitung

³<http://www.naoteamhumboldt.de/de/>

⁴<https://maserati.mi.fu-berlin.de/carolo-cup/>

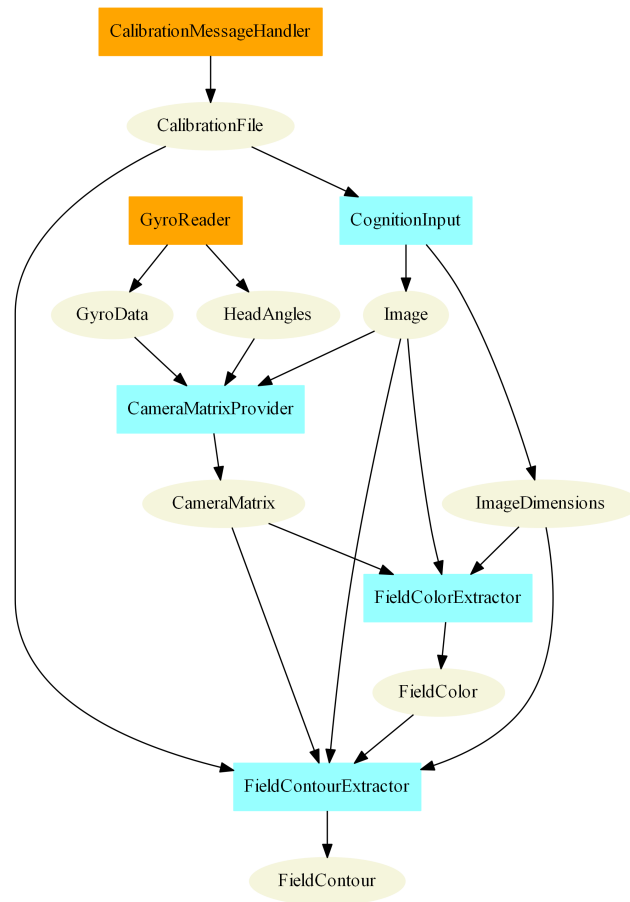


Abbildung 2.5: Graph einer Modulkette zur Ermittlung der Feldkontur (Module rechteckig, Repräsentationen oval)

eines Teammitgliedes wird dadurch auf ein Minimum reduziert und die Einstiegs-
hürde gering gehalten. In der Grafik 2.6 ist der beschriebene Aufbau des Berlin
United-Framework dargestellt, wobei der projektbezogene Teil in Abschnitt „Soccer
Intelligence“ wiederzufinden ist.

2.3 Perzepte und Modelle

Im Rahmen der bestehenden FUsmanoid-Software existieren Perzepte und Modelle
von Objekten. Ein Perzept beschreibt ein durch die Kamera aufgenommenes und
erkanntes Objekt, wie zum Beispiel den Ball oder das Tor. Das Perzept ist nur auf das
aktuelle Bild (Frame) bezogen und beinhaltet hauptsächlich die Objektkoordinaten
sowie die Objektgültigkeit. Ein Perzept ermöglicht also nur eine Aussage über den
gegenwärtigen Zustand eines Objekts.

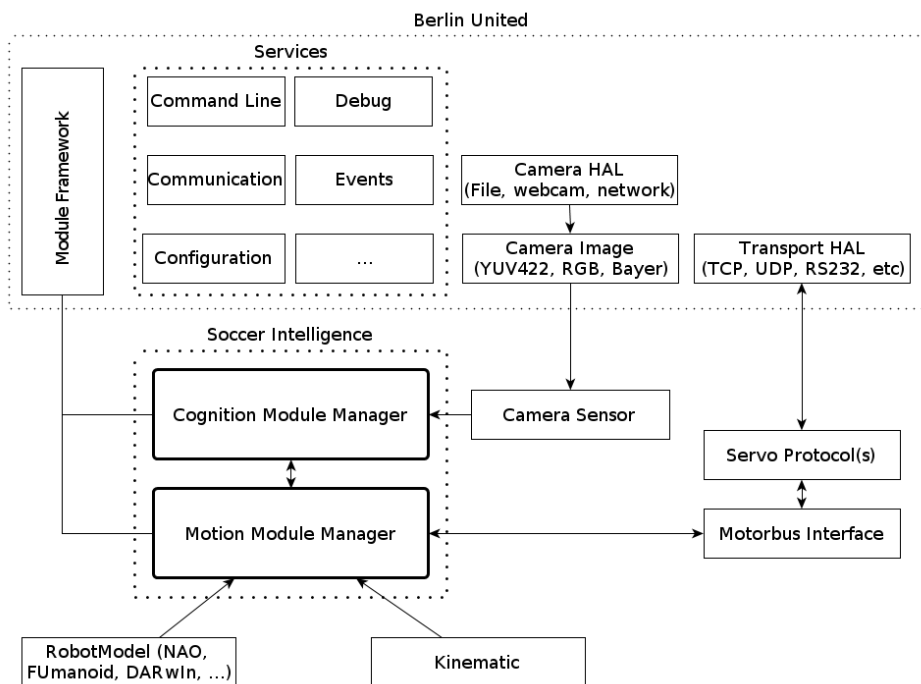


Abbildung 2.6: Aufbau des Berlin United-Frameworks [8]

Hingegen ist ein Modell eine Abstraktion eines Perzepts, bei dem beliebige Parameter verwendet und modelliert werden. Im Gegensatz zum Perzept ist ein Modell nicht nur auf ein Bild bezogen und die Gültigkeit kann beliebig definiert werden. So kann zum Beispiel das Ballmodell erst nach drei Frames, in denen kein Ball erkannt wurde, als ungültig erklärt werden oder aber das Vertrauen in das Modell als eine Zahl wiedergegeben werden, wodurch der Nutzer dieser Information selber entscheiden kann, wann er das Modell für ungültig hält. Liegt der Ball im Zentrum des Kamerabildes ist es nahezu ausgeschlossen, dass sich der Ball im nächsten Frame, was bei 30 Bildern pro Sekunde nur circa 33 Millisekunden dauert, außerhalb des Kamerabildes befindet und dadurch nicht erkannt wird. Wahrscheinlicher ist es, dass die Ballerkennung den Ball in diesem Frame nicht korrekt erkannt hat und der Ball sich somit noch an seiner alten Position oder aber in der Nähe dieser befindet. Zusätzlich kann ein Modell zusätzliche Parameter wie Ballfarbe und Ballgröße beinhalten. Die Definition des Modells ist hierbei dem Entwickler überlassen und ist jeweils an den Verwendungszweck angepasst. Die Modellierung hat dabei den Vorteil, dass über einen gewissen Zeitraum hinweg mehr Informationen aus den einzelnen Frames extrahiert werden können, was zum Beispiel bei der Berechnung der Ballgeschwindigkeit benötigt wird oder aber um die Ballkoordinaten auch außerhalb des sichtbaren Bereichs durch zusätzliche Informationsquellen berechnen zu können.

2.4 Roll-Pitch-Yaw-Winkel

Die Roll-Pitch-Yaw-Winkel finden in dieser Arbeit Verwendung und sollen daher an dieser Stelle näher erläutert werden. Die deutschen Namen der Winkel lauten *Roll* (roll), *Nick* (pitch) und *Gier* (yaw). Der Inertialsensor (IMU) liefert diese Winkel und ist am Kopf des Menschen angebracht, wodurch die Orientierung des Kopfes verfolgt werden kann (Head-Tracking). Die Abbildung 2.7 stellt die Winkel anhand eines Flugzeugmodells dar. Die yaw-Ebene wird beim Schütteln des Kopfes, wie beim „Nein“ sagen, genutzt, stellt also eine Rotation um die z-Achse des Bezugssystems dar. Die pitch-Ebene ist genau das Gegenteil und stellt die Kopfbewegung beim „Ja“ sagen dar. Diese Nickbewegung des Kopfes in der pitch-Ebene ist eine Rotation um die y-Achse. Die roll-Ebene stellt die letzte, verbleibende Kopfbewegung dar, das Neigen des Kopfes zur Seite, also die Annäherung eines Ohrs an die Schultern ist eine Rotation um die x-Achse.

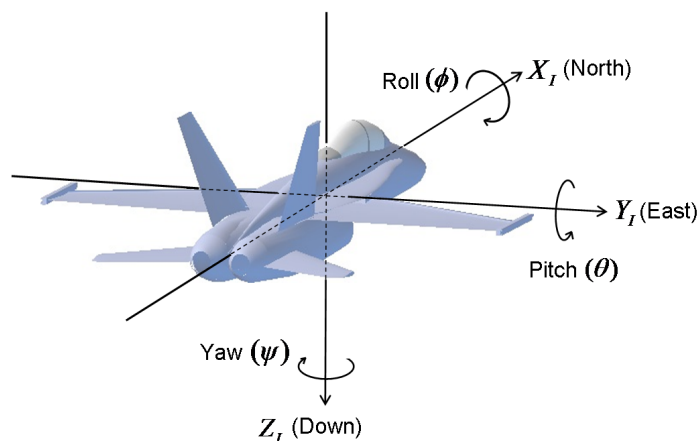


Abbildung 2.7: Veranschaulichung der Roll-Pitch-Yaw-Winkel [9]

2.5 XABSL

„Extensible Agent Behavior Specification Language“ (XABSL) ist eine vereinfachte Programmiersprache zum Verhaltensdesign von autonomen Robotern, welche im Jahr 2004 vom *German Team*⁵, einer Gruppe aus deutschen Universitäten, entwickelt wurde. Hierbei wird das Verhalten anhand eines hierarchischen, endlichen Zustandsautomaten abgebildet, was im Gegensatz zu normalen, endlichen Zustandsautomaten eine überschaubare Zustandsmenge ermöglicht. [10]

⁵<http://www.germanteam.org/>

Ein XABSL-Verhalten wird mit den folgenden fünf Grundbausteinen gestaltet:

Agenten Agenten stehen für ein gewünschtes Verhalten des Roboters, wie zum Beispiel „spiele Fußball ohne Schiedsrichter“ oder „schieße einen Elfmeter“. Ein Agent besteht aus mehreren Optionen, welche in einem azyklischen Graphen sortiert sind. Hierbei wird bei einer definierten Option (Wurzel) mit der Abarbeitung begonnen.

Optionen Eine Option ist ein endlicher Zustandsautomat und steht für einzelne Teilaufgaben, wie zum Beispiel „gehe zum Ball“ und „dribble den Ball“. Die einzelnen Optionen sind hierarchisch strukturiert um komplexe Problemstellungen in Teile aufzubrechen und einzeln zu bearbeiten. So kann zum Beispiel die Option „dribble den Ball“ in einem seiner Zustände die primitivere Unteroption „rotiere um Grad x “ aufrufen, um die benötigte Startposition einzunehmen. Eine Option besteht aus mehreren Zuständen.

Zustände Zustände führen bestimmte Aktionen aus und beinhalten implizit Informationen über zeitliche Abfolgen. So kann der Zustand „Aufstehen“ eine Aufstehbewegung auslösen, was eine vorherige Detektion eines Sturzes bedingt, was wiederum durch definierte IMU-Werte ermittelt wird.

Entscheidungsbaume Jeder Zustand besitzt einen Entscheidungsbaum. Mit ihm wird ermittelt, in welchen Folgezustand sich der Zustandsautomat begibt. Als Entscheidungsgrundlage stehen Symbole zur Verfügung, welche eine Schnittstelle zum externen Code darstellen.

Symbole Ein Symbol kann ein boolescher, ein numerischer oder ein aufzählbarer Wert sein. Zum Beispiel kann der Zustand „stabiles Aufstehen“ nur erreicht werden, falls vorher die IMU-Werte eine Änderung des Symbols (aufzählbar) „robot_pose“ von „standing“ auf „fallen“ bewirkt haben, der liegende Zustand aus Stabilitätsgründen mindestens 300 Millisekunden beibehalten wurde (Symbol „state_time“, numerisch) und der Bewegungsmodifikator (Symbol „stable_movements“, boolesch) positiv ausfiel.

Diese Grundbausteine werden beim Verhaltensdesign genutzt, um basierend auf externen Ereignissen (Kamera und IMU) in entsprechende Zustände zu wechseln, welche unterschiedliche Befehle auslösen. Zusätzlich zu den bereits vorhandenen XABSL-Symbolen, wie zum Beispiel Informationen über das Tor oder den Ball, wurden für die Arbeit neue Symbole eingefügt, welche zur Interaktion mit dem Menschen dienen. Nachfolgend ist im Bild 2.8 eine einfache Option visualisiert, welche bei der RoboCup-Demonstration die Bewegungsrichtung des Menschen steuerte. Die Zustände sind als Kreise dargestellt und die Pfeile geben die Übergänge der Entscheidungsbaume wieder.

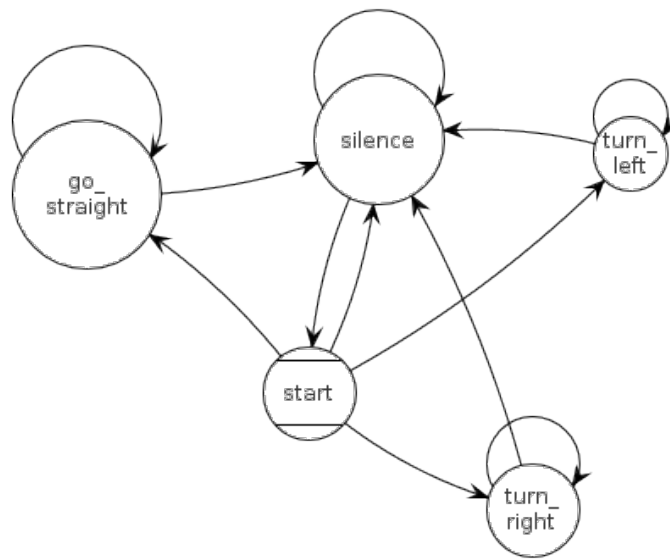


Abbildung 2.8: Darstellung einer XABSL-Option

2.6 OpenAL

Da für den Direct Mode eine möglichst reale, akustische 3D-Umgebung generiert werden sollte, lag es nicht fern eine Audio-API zu wählen, welche vermehrt Verwendung in Computerspielen findet, in denen, neben der Optik, auch die Akustik eine große Rolle spielt. Für den Command Mode hingegen ist eine einfache Ausgabe von Audiosignalen ausreichend, allerdings können die zusätzlichen Funktionalitäten für eine verbesserte Mensch-Maschine-Kommunikation verwendet werden. Daher fiel die Wahl der Audio-API auf OpenAL [11], eine standardisierte Audio-API, welche durch die Industrie im Jahr 2000 in der Version 1.0 veröffentlicht wurde und eine Schnittstelle zur Soundarchitektur des Betriebssystems darstellt. Unter Linux werden unter anderem „Advanced Linux Sound Architecture“ (ALSA) und „PulseAudio“ unterstützt. OpenAL ist plattformübergreifend und simuliert Tonquellen und Tonempfänger in einem dreidimensionalen Raum, was ein Pendant zur graphischen API „OpenGL“ darstellen soll. Im Rahmen dieser Masterarbeit kam OpenAL Soft 1.13⁶ zum Einsatz, eine Implementation der OpenAL-API, welche unter „GNU Lesser General Public License“ (LGPL) veröffentlicht wurde.

Für die Nutzung von OpenAL sind folgende Schritte zu beachten: Der erste Schritt zur Erzeugung von Tönen ist die Definition des gewünschten Audiogerätes des Betriebssystems, welches für die Ausgabe des Tons genutzt werden soll. Im Rahmen dieser Arbeit wurde immer das Standard-Audiogerät des Betriebssystems genutzt,

⁶<http://kcat.strangesoft.net/openal.html>

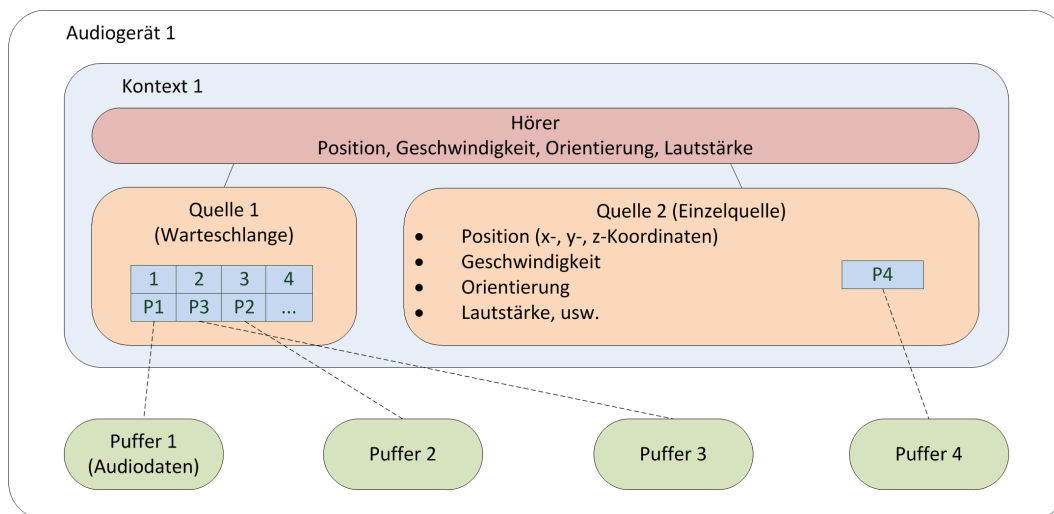


Abbildung 2.9: Grundobjekte OpenAL

was im Großteil aller Fälle die richtige Wahl sein sollte.

Wie in der Grafik 2.9 zu erkennen, wird als nächstes ein Kontext benötigt, welcher an das Audiogerät gebunden wird. Kontexte können mehrfach existieren und dienen zum schnellen Szenewechsel. Außerhalb vom Kontext, aber innerhalb des Audiogerätes, gibt es Audiopuffer, welche unkomprimierte Audiodaten beinhalten. Als speicherintensivster Teil der Audiowiedergabe sind diese Audiopuffer kontextübergreifend, sodass die gleichen Audiodaten in mehreren Kontexten genutzt werden können, ohne multiple Puffer mit redundanten Daten füllen zu müssen.

Innerhalb eines Kontextes findet nun die Definition von Tonquellen und Tonempfänger (Hörer) statt, wobei der Letztere nur einmal vorhanden ist. Er repräsentiert die Ohren des Zuhörers im dreidimensionalen Raum, welcher durch verschiedene Tonquellen beschallt werden kann. Beide Objekttypen haben für die Simulation im 3D-Raum eine Position, eine Orientierung und eine Lautstärke. Die Geschwindigkeit, auf dessen Grundlage der Dopplereffekt berechnet wird, sind bei den zu erwartenden Situationen nicht relevant.

Damit eine Tonquelle einen Einfluss auf das Hörereignis hat, muss diese mit einem Audiopuffer verbunden werden, da dieser die nötigen Audiodaten enthält. OpenAL unterstützt hierbei zwei verschiedene Tonquellen, eine Warteschlange und eine Einzelquelle. Mit der Einzelquelle wird nur ein einziger Puffer verknüpft, welche nachfolgend abgespielt, pausiert oder gestoppt werden kann. Erreicht der Puffer der Einzelquelle sein Ende, wird das Abspielen gestoppt oder wahlweise vom Anfang an neu begonnen. Im Gegensatz dazu kann die Warteschlange mit einer Vielzahl von Puffern verbunden werden. Beginnend mit dem ersten Puffer beginnt die Abarbeitung und geht beim Erreichen des Endes zum folgenden Puffer über. Das Abspielen endet erst,

falls kein Folgepuffer mehr existiert. Durch rechtzeitiges Hinzufügen von neuen Puffern kann also eine beliebig lange Audiofolge entstehen. Die gleichzeitige Verwendung des gleichen Puffers für mehrere Tonquellen oder sogar die mehrfache Verwendung innerhalb einer Warteschlange wird durch OpenAL unterstützt. [12]

KAPITEL 3

Ähnliche Arbeiten

3.1 3D-Audio im Flugverkehr

Eine vielversprechende Anwendungsmöglichkeit von dreidimensionalen Audio-Informationen bietet der allgemeine sowie militärische Flugverkehr. In der Masterarbeit von Jacque M. Joffrion [13] wird der Informationsgewinn durch zusätzliche 3D-Audio-Informationen im Cockpit eines Flugzeuges näher untersucht. Es ist leicht nachvollziehbar, dass gerade bei beeinträchtigter Sicht eine richtungsgenaue Wiedergabe von akustischen Informationen einen Mehrertrag zu den üblichen Instrumenten des Cockpits darstellt. So könnte zum Beispiel eine Durchsage vom Tower aus eben der Richtung des Towers eine bessere Orientierung ermöglichen, da gerade diese bei Nachtflügen oder schlechten Wetterverhältnissen erschwert ist. Auch könnte eine Kollisionswarnung aus der entsprechenden Richtung kommen, was die Reaktionszeit entscheidend verbessern könnte. Damit der Blick eines Piloten für wichtige Informationen nicht auf die stationären Instrumente gerichtet werden muss, existieren bereits Head-up-Displays, welche allerdings keine korrekten Richtungsinformationen vermitteln können, da die Kopfposition des Piloten unbekannt ist.

In benannter Arbeit wurde die Kopfposition des Piloten mit Hilfe einer IMU (Head-Tracker) bestimmt, um eine Anpassung der Audio-Informationen zu ermöglichen. Dabei wurden in unterschiedlichen Experimenten die Erkennungshäufigkeiten der Tonrichtungen ermittelt und ausgewertet. Als Referenz diente die Tonortung, welche sich an der Ausrichtung des Flugzeuges orientierte, also sich die Audio-Informationen nicht anhand der Kopfbewegungen anpassten. Für die Ortung der Höhe einer Tonquelle, also der Elevation (siehe Abschnitt 2.3), ergab sich keine signifikante Verbesserung gegenüber der Referenz, was bedeutet, dass auf dieser Achse kaum zusätzliche Informationen übermittelt werden können. Jedoch ergab sich auf der Horizontalebene, dem Azimut, ein besseres Ergebnis. Hier musste entsprechend einer analogen Uhr die Richtung des Tons angegeben werden, also zwölf mögliche Ant-



Abbildung 3.1: Pilot mit einer IMU an das Headset montiert (Kasten an der Oberseite des Headsets). [13]

worten. Die Anzahl der korrekten Antworten erhöhte sich gegenüber der Referenz, welche bei 40 Prozent lag, auf über 70 Prozent, was auch daran lag, dass die in Abschnitt 2.1.2 beschriebenen Mehrdeutigkeiten durch die Kopfbewegungen des Piloten aufgelöst werden konnten. Viele Faktoren bei den Experimenten waren nicht optimal, wie zum Beispiel die Genauigkeit der IMU oder hohe Latenzen durch das verwendete Betriebssystem, wodurch sich weiteres Verbesserungspotenzial für die Zukunft ergibt.

3.2 3D-Audio-Schnittstelle für Blinde

Eine andere Arbeit [14] beschäftigt sich mit der Verbesserung von Computerschnittstellen für Personen mit Sehbehinderung. Dabei werden ebenfalls 3D-Audio-Informationen als zusätzlicher Informationskanal genutzt, da der Hörsinn ein natürlicher Ersatz für den fehlenden Sinn darstellt und bei den Betroffenen besonders ausgeprägt ist. Mit der Brailleschrift wird bereits die taktile Wahrnehmung als Informationskanal genutzt, welcher allerdings nur eine geringe Bandbreite für Informationen bietet. Die Autoren haben einen dreidimensionalen, virtuellen Raum entworfen, welcher mit Hilfe eines Joysticks durch den Hörsinn erkundet werden kann. Dieser Raum soll ein Äquivalent zum normalem Desktop darstellen. In diesem befinden sich „Earcons“, welche Icons repräsentieren sollen, an definierbaren Positionen mit individuellen Erkennungstönen. Der Nutzer kann sich zum benötigten „Earcons“ begeben und die übliche Aktionen des Betriebssystems ausführen. Die Möglichkeiten beschränken sich hierbei nicht nur auf den Desktop, da die Autoren für mögliche Erweiterungen eine Programmierschnittstelle entworfen und zur Verfügung gestellt haben, womit zum Beispiel die Repräsentation einer Menüleiste möglich wäre.

Wie in der vorherigen Arbeit wird auch hier ein Head-Tracker genutzt, um die Mehrdeutigkeiten der akustischen Repräsentation aufzulösen und somit eine verbesserte Orientierung im virtuellen Raum zu ermöglichen. Der Prototyp des Systems aus dem Jahr 2003 kostete 6000 US-Dollar und benötigte spezielle Hardware. Wie in der

Arbeit vorhergesehen wurde, benötigt man mit der heutigen Rechenleistung keine spezielle Hardware zur Konstruktion einer solchen virtuellen Umgebung, wodurch sich die Kosten für ein solches System enorm reduzieren. Setzt man einen Computer und Kopfhörer voraus, dann bleibt die IMU die einzige zusätzliche Hardware, welche heute USB-kompatibel für circa 100 US-Dollar erhältlich ist.

3.3 3D-Audio für erweiterte Realität

Ein ähnliches System, wie in dieser Arbeit, wurde auch in folgender Arbeit [15] entworfen. Dieses System bietet jedoch mehr Sensoren, wie zum Beispiel GPS, und ist verstärkt auf die Kommunikation mit Dritten ausgelegt, was an der „Voice over IP“-Unterstützung (VOIP) sowie der Push-to-Talk-Funktion erkennbar ist. Dieses militärische System nutzt die zusätzlichen Audio-Informationen zum Orten von Gefahrenquellen und zum Navigieren zu Zielpunkten. Die Kopfposition wird auch hier mit einem Head-Tracker gemessen und für die virtuelle Repräsentation genutzt. Im Rahmen eines Experiments wurden Personen durch zusätzliche, visuelle Informationen an die akustischen Repräsentationen gewöhnt und die Ortungsgenauigkeit über die Zeit beobachtet. Durch dieses Training stieg hauptsächlich die Genauigkeit bei der Feststellung, ob sich ein Objekt vor oder hinter der Person befand.



Abbildung 3.2: Tragbares System mit erweiterter Realität durch 3D-Audio-Informationen [15]

3.4 Oculus Rift

Auch im kommerziellen Bereich gibt es erneut Ansätze eine virtuelle Realität für den Menschen zu erschaffen und den Menschen in eben diese eintauchen zu lassen. Frühere Modelle einer Videobrille, wie zum Beispiel der Virtual Boy von Nintendo¹, konnten sich aufgrund multipler Einschränkungen nicht durchsetzen. Ein neues Modell einer solchen Virtual-Reality-Brille ist die Oculus Rift², welche sich gerade in der Entwicklung befindet und viel Aufmerksamkeit seitens der Presse, aber auch der Nutzer, erfährt. Die Oculus Rift verfolgt nicht, wie die vorhergehenden Arbeiten, das Ziel die Realität zu erweitern, sondern diese zu ersetzen. Die Sicht des Menschen wird durch eine Videobrille nahezu komplett ersetzt, was eine starke Einbindung des Menschen in die virtuelle Umgebung erzeugt. Zu diesem Zweck ermittelt ein Head-Tracker die Orientierung des Kopfes und nutzt diese zur Erzeugung der virtuellen Realität.



Abbildung 3.3: Developer-Version der Oculus Rift ³

¹<http://www.nintendo.co.jp/n09/vue/index.html>

²<http://www.oculusvr.com/>

³http://commons.wikimedia.org/wiki/File:Orlovsky_and_Oculus_Rift.jpg, CC BY-SA 2.0

KAPITEL 4

Hardware

4.1 Prototyp 1 und 2



Abbildung 4.1: Erster Prototyp mit Kamera (Bildunterseite) und IMU (rot) auf einen Fahrradhelm montiert.

Der erste Prototyp wurde gemäß dem vorgestellten Konzept [16] entworfen und beinhaltete die gleiche elektrische Hardware wie die humanoiden Roboter der FUmoids (2012). Da der Mensch jegliche Aktionen ausführt, konnte auf alle Motoren verzichtet werden. Um die Sensoren am Kopf des Probanden zu befestigen, wurde ein Fahrradhelm genutzt, auf den Kamera sowie IMU montiert wurden. Die Größe der Sensoren bedingte einer stabilen Befestigung, wodurch eine geeignete Plattform gewählt werden musste, welche bequem am Kopf eines Menschen befestigt werden kann. Aus Kostengründen setzten sich hierbei Fahrradhelme gegen zum Beispiel Motorradhelme durch. Die restliche Hardware wurde zum besseren Transport in einer Bauchtasche untergebracht. Der Nachteil dieser Version ergab sich durch die ver-

stellbare Kamera, welche durch die Bewegungen während der Nutzung ihre Ausrichtung verlieren konnte. Eine veränderte Ausrichtung der Kamera führte zu einem geänderten, undefinierten Winkel gegenüber der IMU und somit zu einer erhöhten Ungenauigkeit der Software. Zudem ergab sich mit dem Fahrradhelm ein Größenproblem, da jeder Mensch einen unterschiedlich großen Kopf besitzt und demnach einen passenden Fahrradhelm benötigt, um einen komfortablen sowie sicheren Sitz zu ermöglichen. Die Befestigung der Kamera oberhalb der Augen verlangte eine zusätzliche Abstraktion seitens des Nutzers, da zum Beispiel das Schauen auf die eigenen Füße eine unnatürliche Kopfposition benötigte, welche sich durch die unterschiedlichen Blickwinkel von Kamera und Mensch ergab.

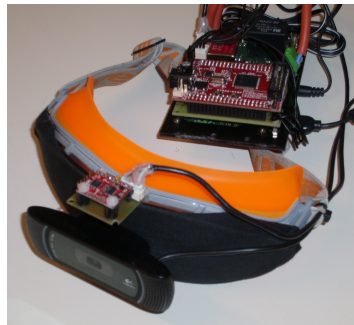


Abbildung 4.2: Zweiter Prototyp mit Kamera und IMU auf eine Arbeitsschutzbrille montiert.

In der zweiten Version wurde das Kameramodell gewechselt, wodurch eine Winkeländerung während der Nutzung nicht mehr möglich war und zugleich die identische Kamera wie die FHumanoids in der Version 2013 verwendet. Die kleinere Hardware ermöglichte eine veränderte Befestigung und somit eine alternative Plattform. Diese neue Plattform sollte möglichst jedem Menschen passen sowie eine feste Anbringung der Sensoren ermöglichen. Eine Kombination mit der benötigten Dunkelbrille schien nicht abwegig, benötigte jedoch eine Brillenform, welche das komplette Gesichtsfeld des Menschen abdeckt und zudem eine ausreichende Stabilität garantiert, um die zusätzlichen Sensoren zu befestigen. Diese Bedingungen wurden durch Ski-Brillen und Arbeitsschutzbrillen erfüllt, wobei sich Letztere durchsetzten, da dort keine unterschiedlichen Größen vorhanden waren, was eine universelle Passform impliziert. Zudem sind Arbeitsschutzbrillen über einen längeren Zeitraum in gleicher Ausführung erhältlich, kostengünstiger sowie in größerer Stückzahl erhältlich. Die stabile Kunststoffscheibe einer solchen Brille ermöglichte die Befestigung der Kamera an eben dieser, wodurch die Kameraposition der natürlichen Position der Augen entsprach. Hierdurch wurde die Ergonomie des Prototypen verbessert und unnötige Abstraktionen verhindert. Die feste Anbringung der Kamera und der IMU an der Brille ermöglichte eine genaue Messung des Winkelunterschiedes und führte dadurch zu einer verbesserten Projektionsgenauigkeit.

Die folgende Hardware wurde in beiden Prototypen verbaut:

- Kamera - Logitech QuickCam Pro 9000 bzw. später Logitech C910 HD Webcam
- Fahrradhelm bzw. später Arbeitsschutzbrille
- Minicomputer - IGEPv2
- Spannungsversorgungsplatine (Eigenbau der FUmoids)
- Akkumulator
- Orientierungssensor (Eigenbau der FUmoids)
- Dunkelbrille

4.2 RoboCup-Prototyp

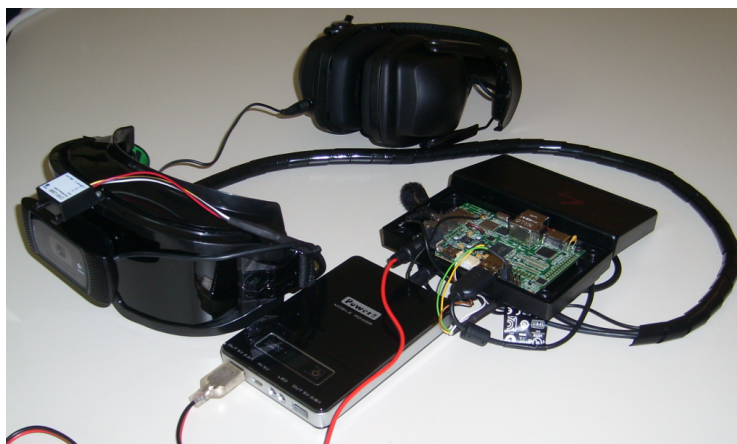


Abbildung 4.3: RoboCup-Prototyp mit allen Komponenten. Kamera mit IMU (links), IGEPv2 (rechts), Akkumulator (unten) und Kopfhörer (oben)

Die Fördermittel ermöglichten eine neue Zusammenstellung der Hardware, welche losgelöst von vorheriger Roboterhardware, besser auf die Bedürfnisse des Projekts abgestimmt wurde. Da mehr als ein Prototyp für die Demonstration benötigt wurde, erfolgte die Auswahl der Hardware auch nach Gesichtspunkten einer einfachen Montage, wodurch Hardwarekomponenten für den RoboCup-Prototypen ausgewählt wurden, welche ohne größere Modifikationen verbaut werden konnten.

4.2.1 Kamera



Abbildung 4.4: Abbildung der Kamera B910¹

Logitech B910 HD Webcam Diese Kamera ² ist baugleich mit dem Modell C910, welche von dem FUmanoids verwendet wird. Sie unterstützt den aktuell verwendeten Aufnahmemodus von 640 x 480 Pixeln bei 30 Bildern pro Sekunde, ist aber auch für eine spätere Auflösungserhöhung geeignet, da ebenfalls 1280 x 720 Pixel mit 30 Bildern pro Sekunde unterstützt werden. Durch die Unterstützung in der FUmanoid-Software ist eine Konfiguration vieler Kameraparameter wie Kontrast, Sättigung, Helligkeit usw. möglich, welche für jedes Spielfeld individuell angepasst werden müssen.

4.2.2 Minicomputer



Abbildung 4.5: Abbildung des Minicomputers [17]

IGEPv2 Der IGEPv2 [17] ist ein kleiner Computer mit ARM Cortex A8 1GHz CPU und 512 MB RAM, auf welchem sämtliche Datenverarbeitungen stattfinden. Mit umfangreichen Anbindungsmöglichkeiten können alle nötigen Peripheriegeräte angeschlossen werden, wofür Ethernet-, USB-, HDMI- und UART-Anschlüsse vorhanden sind. Für drahtlose Übertragungen stehen Wifi 802.11 b/g und Bluetooth 2.0 zur Verfügung. Zusätzlich zum internem Flashspeicher von 512 MB dient ein

¹<http://www.logitech.com/assets/33038/logitech-b910-hd-webcam-gallery-3.png>

²<http://www.logitech.com/de-de/product/b910-hd-webcam>

MicroSD-Kartensteckplatz als Datenspeicher, welcher in diesem Projekt mit einer 8 GB MicroSD-Karte ausgestattet war. Der Anschluss von Audiogeräten erfolgt über 3,5 mm Stereo-Klinkenstecker, welche zur Ausgabe und Aufnahme dienen.

4.2.3 Akkumulator



Abbildung 4.6: Abbildung des Akkumulators³

Vitebo VTB 22 Dieser Akkumulator [18] dient zur Stromversorgung aller elektronischen Bauteile. Mit einer Kapazität von 7200 mAh ist die Funktionsfähigkeit für mehrere Stunden sichergestellt, womit mehrere Spiele überbrückt werden können. Eine digitale Anzeige hilft beim Ablesen des aktuellen Ladestandes in Prozent. Weiterhin sind zwei USB-Anschlüsse vorhanden, einer mit maximal 2,2 Ampere Ausgangsleistung und einer mit 0,8 Ampere Ausgangsleistung. Die angegebenen Spannungen von 5 Volt bis 5,5 Volt wurden messtechnisch überprüft, da der IGEPv2 nur für eine maximale Eingangsspannung von 5,2 Volt ausgelegt ist. Diese Grenze wurde nur bei dem 2,2 Ampere Anschluss eingehalten. Um eine Beschädigung vom IGEPv2 zu vermeiden, darf also nur der 2,2 Ampere Anschluss verwendet werden. Die Aktivierung der Stromversorgung erfolgt durch einen Knopfdruck auf dem Akkumulator und dem Einstecken der IGEPv2-Stromversorgung, welche in Form eines USB-Steckers angefertigt wurde. Die Deaktivierung jeglicher Hardware erfolgt durch das Ziehen des besagten USB-Steckers. Die Stromversorgung der Sensoren erfolgt über unterschiedliche Anschlüsse des IGEPv2.

³<http://www.reichelt.de/Mobile-Powerpacks/VTB-22/3/index.html?;ARTICLE=117929>;

4.2.4 Inertialsensor (IMU)



Abbildung 4.7: Abbildung des Inertialsensors⁴

CH Robotics UM6 Dieser Orientierungssensor [19] verfügt über neun Freiheitsgrade, da Gyroskop, Beschleunigungssensor und Magnetometer mit jeweils drei Achsen zur Ermittlung der aktuellen Orientierung genutzt werden. Der UM6 bietet die Möglichkeit, die gewünschten Daten mit einer konfigurierbaren Frequenz über die zwei Kommunikationskanäle zu übermitteln. Die Kommunikation erfolgt über eine TTL UART (3,3 Volt) oder einem SPI Bus, wobei für dieses Projekt die Anbindung an den IGEPv2 mittels UART stattfand. Der UM6 bietet viele Möglichkeiten der automatischen sowie manuellen Fehlerkompensation, welche durch spezielle Befehle beeinflusst werden können. Eine Kalibrierung der IMU wird in der Anleitung empfohlen, sobald sich das Modul am Montageort befindet und somit der Einfluss der Umgebung mit in die Kalibrierungswerte einfließt. Diese Kalibrierung wurde mit dem bereitgestellten Programm des Herstellers durchgeführt, wofür jedoch ein Microsoft-Betriebssystem nötig war.

Die IMU ist neben der Kamera der einzige Sensor und stellt eine wichtige Informationsquelle über die aktuelle Ausrichtung des Kopfes sowie der Kamera dar. Eine Überprüfung der Genauigkeit dieser Informationsquelle ist nötig, da fehlerhafte Daten einen negativen Effekt auf mehrere Softwaremodule haben, was wiederum spezielle Anpassungen auf Softwareebene nötig machen würde oder sogar die Nutzung der IMU ausschließen würde. Um die Fehler während einer Bewegung zu messen, müsste ein externes Messinstrument herangezogen werden, welches die exakte Ausrichtung des Moduls zeitlich und räumlich sehr genau erfasst, um so einen Vergleich mit den internen Messwerten zu ermöglichen. Auf eine solch aufwändige Auswertung wurde jedoch verzichtet und stattdessen mit den vorhandenen Mitteln gearbeitet, um einen indirekten Rückschluss auf die Genauigkeit zu ziehen.

Für die erste Messung wurde das Kopfmodul in allen Richtungen gedreht und anschließend auf eine stabile Unterlage gelegt, wobei der Zeitpunkt des Hinlegens in den Daten per Tastendruck markiert wurde und die Startposition unterschiedlich gegenüber der Endposition war. Hierdurch kann die Zeit gemessen werden, welche die IMU benötigt, um einen statischen Wert nach einer vorherigen Bewegung anzunehmen.

⁴<https://www.chrobotics.com/shop/orientation-sensor-um6>

Auch die Stärke der Winkeländerungen spielen in diesem Zeitraum eine Rolle. In der Abbildung 4.8 ist eine dieser Messungen exemplarisch dargestellt, in welcher sich die Bewegungsphase sowie Abklingphase erkennen lässt. Der Zeitpunkt des Hinlegens ist mit einer vertikalen Linie markiert und die Abklingphase ist in einem separatem Graph vergrößert dargestellt. In der Abklingphase lässt sich erkennen, dass bereits zum Zeitpunkt des Hinlegens nur kleine Winkelabweichungen zu den endgültigen Achsenwinkeln bestehen und dass die Annäherung an einen statischen Wert nur wenige Sekunden dauert. Diese Erkenntnis wurde durch mehrere Messungen bestätigt und lässt den Schluss zu, dass auch während der Bewegungen des Kopfmoduls nur ein geringer Winkelfehler besteht, da der Mensch in diesem Szenario keine heftigen Bewegungen durchführt.

Zudem wurde eine Messung der Achsenwerte über einen längeren Zeitraum durchgeführt, um ein eventuelles Driftverhalten festzustellen, wovon hauptsächlich der Yaw-Winkel betroffen wäre, da dieser überwiegend durch das trägere Magnetometer bestimmt wird. Der Yaw-Winkel bietet eine gute Möglichkeit die horizontale Orientierung auf dem Spielfeld zu bestimmen und müsste gesondert korrigiert werden, falls dieser einen Drift aufweist. Während der Messung wurde das Kopfmodul alle zwei Minuten an einer definierten Nullposition fixiert und nach einer zehnssekündigen Abklingphase die aktuellen Messwerte notiert. Zwischen diesen Messphasen wurde das Modul zufällig in allen Dimensionen rotiert, um die Bewegungen während eines Spiels zu simulieren. Im Graph 4.9 sind die Abweichungen der Achsen zur Nullposition über einen Zeitraum von 30 Minuten dargestellt, was zeitlich mehr als zwei Spielhälften abdeckt. Der Yaw-Winkel zeigt über diesen Zeitraum nur einen Drift von drei Grad, was für den Verwendungszweck absolut ausreichend ist. Durch einen höheren Drift hätte es zum Beispiel zu der Verwechslung der Tore kommen können, da diese sich optisch nicht unterscheiden und somit über die horizontale Ausrichtung zugeordnet werden müssen, was bei einem hohen Drift mit zunehmender Zeit immer unsicherer werden würde und letztendlich zu einer Vertauschung der Tore führen könnte.

4.2.5 Pegelwandler

Sparkfun RS232 Shifter SMD Der Pegelwandler [20] wird für die Anbindung des Inertialsensors benötigt, da die Kommunikationsarten vom Sensor und dem UART-Anschluss auf dem IGEPv2 nicht übereinstimmen. Der UART-Anschluss des IGEPv2 arbeitet nach dem RS-232-Standard, während die IMU mit 3,3 Volt TTL-Pegeln operiert. Beim RS-232-Standard erfolgt die Signalisierung einer logischen Eins durch eine negative Spannung und eine logische Null durch eine positive Spannung, wohingegen die TTL-Pegel der IMU nur zwischen positiven Spannungen wechseln. Der

⁵http://www.lipoly.de/index.php?main_page=product_info&products_id=104471

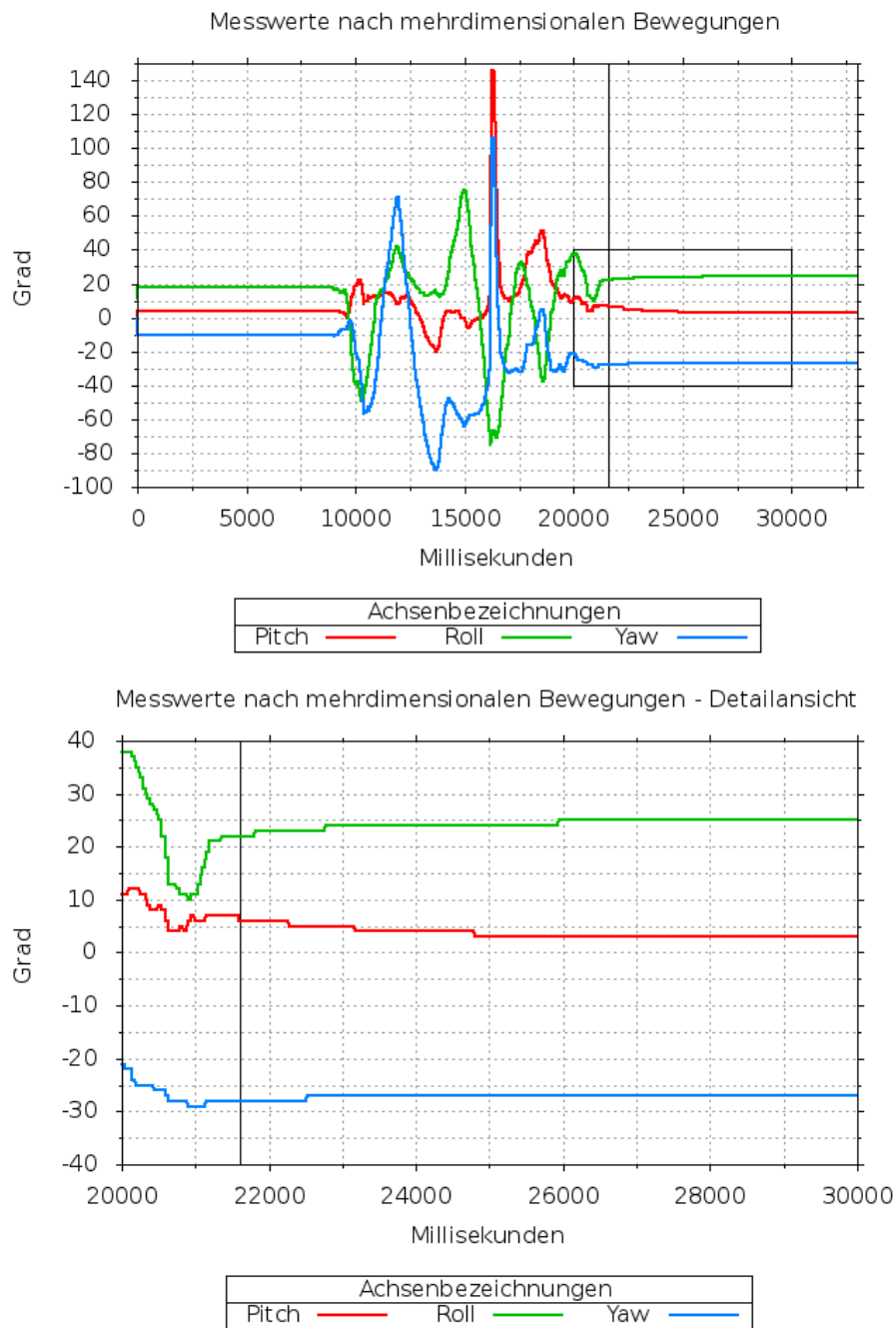


Abbildung 4.8: Anpassungsdauer und Winkelabweichungen der IMU

Pegelwandler konvertiert die entsprechenden Spannungspegel, sodass eine Kommunikation möglich wird. Die Stromversorgung des Pegelwandlers erfolgt über einen gesonderten Anschluss auf dem IGEPv2, welcher eine Betriebsspannung von 5 Volt

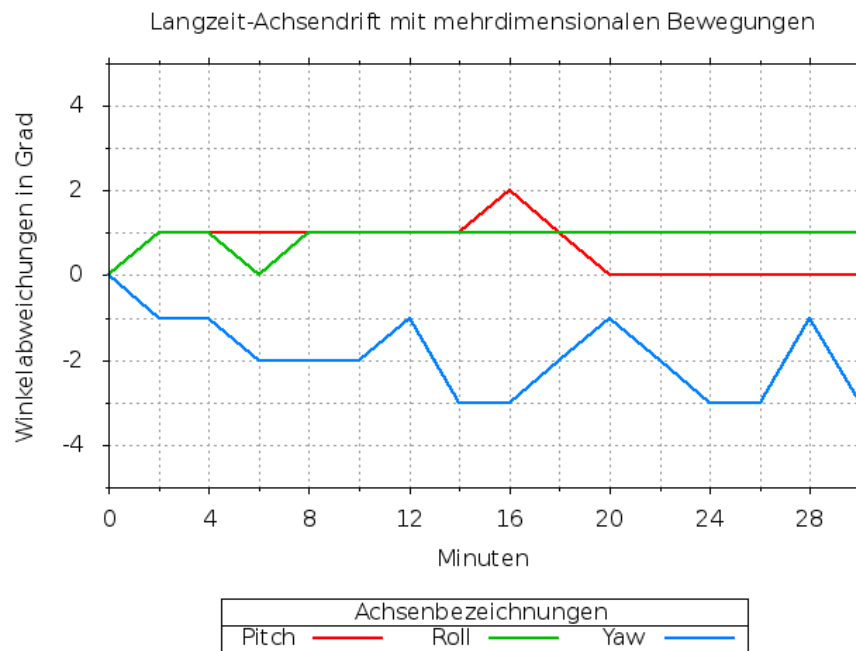
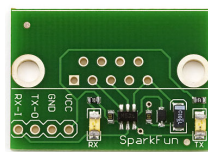


Abbildung 4.9: Langzeitdrift der IMU

Abbildung 4.10: Abbildung des Pegelwandlers⁵

zur Verfügung stellt. Bedingt durch den internen Aufbau des Pegelwandlers musste eine zusätzliche Diode zur Reduzierung der Betriebsspannung eingesetzt werden, da ansonsten ein ständig leitender Transistor eine Kommunikationsrichtung blockierte.

4.2.6 Arbeitsschutzbrille

UVEX ultravision 9301 Bei diesem Modell einer Arbeitsschutzbrille handelt es sich um eine Schweißervollsichtbrille [21], welche auch für Brillenträger geeignet ist. Durch die Abschirmung der Augen in alle Richtungen ergibt sich der gewünschte Sichtschutz. Die Sichtscheibe, welche nicht komplett undurchsichtig ist, wurde im Nachhinein von innen mit schwarzer Farbe undurchsichtig gemacht. Für die Befestigung der Kamera wurde ein Loch in die horizontale sowie vertikale Mitte der Sichtscheibe gefräst und die Kamera mit einem Splint fixiert. Das Fräsen und das Auf-

⁵http://www.mercateo.com/p/281-402428862/UVEX_Vollbrille_9301_245_ultravision_schwarz.html



Abbildung 4.11: Abbildung der Arbeitsschutzbrille⁶

bringen der Farbe wurden durch die herausnehmbare Sichtscheibe erleichtert. Eine Anpassung an unterschiedliche Kopfgrößen erfolgt durch ein verstellbares Kopfband.

4.2.7 Gehörschutzkopfhörer



Abbildung 4.12: Abbildung des Gehörschutzkopfhörers⁷

Howard Leight SYNC Diese Gehörschutzkopfhörer⁸ dienen zur auditiven Übermittlung von Informationen an den Menschen. Statt beliebige Kopfhörer zu nutzen, wurden diese Kopfhörer erworben, da diese durch ihre geräuschkämmende Wirkung vor Störquellen schützen und eine Beeinflussung durch die Außenwelt verhindern sollen. Zudem wird durch die Form und das Material des Kopfhörers eine hygienische Nutzung durch mehrere Personen möglich, da der Kopfhörer über den Ohren liegt und abwischbar ist. Eine Nutzung personenbezogener, individueller Kopfhörer, um so hygienischen Ansprüchen gerecht zu werden, war durch die Schutzhülle des IGEPv2 nicht möglich, welche keinen problemlosen Wechsel der Kopfhörer ermöglichte. Auch hätte die Nutzung von heterogenen Kopfhörern zu unterschiedlichen Höreindrücken

⁷http://g-ecx.images-amazon.com/images/G/01/th/content_26/Q2_2011/c26-B004U4A5RU-1-l.jpg

⁸<http://www.howardleight.com/ear-muffs/sync>

führen können, was unnötige Anpassungen auf Softwareebene zur Folge gehabt hätte.

4.3 Aufbau

In Abbildung 4.13 ist der schematische Aufbau des RoboCup-Prototypen dargestellt.

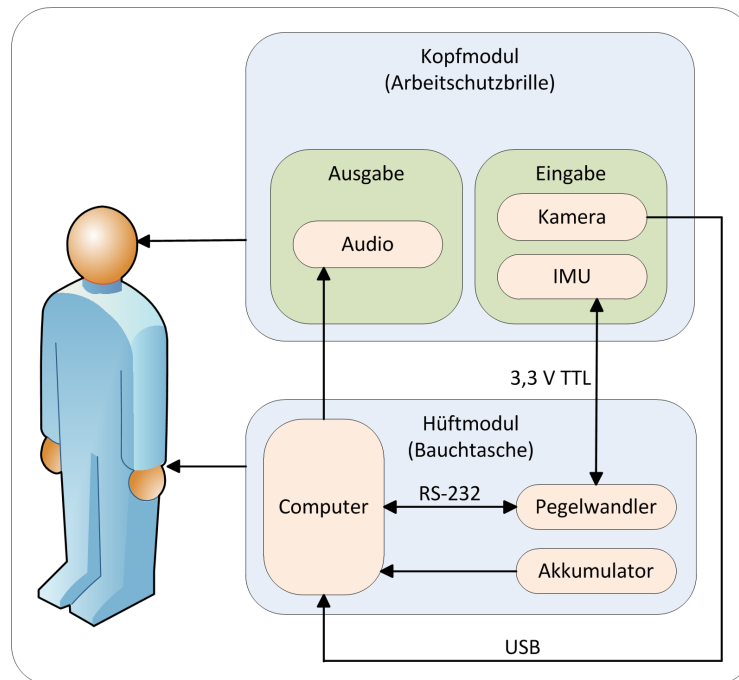


Abbildung 4.13: Grundaufbau des RoboCup-Prototypen

Wie im Bild des RoboCup-Prototypen (siehe Abbildung 4.3) zu erkennen, ist die IMU an der oberen Vorderkante der Arbeitsschutzbrille angebracht. Eine stabilere Montage an den Gehörschutzkopfhörer wurde verworfen, da ein erhöhter Kalibrierungsaufwand vonnöten gewesen wäre und die Gefahr einer Winkeländerung während der Nutzung nicht ausgeschlossen werden konnte. Wie die Verbesserung zum zweiten Prototyp, ermöglicht die fest definierte Position der IMU in Bezug auf die Kamera eine erhöhte Genauigkeit der Projektion auf Softwareebene.

Für die serielle Kommunikation mit der IMU wurde die UART-Schnittstelle (Universal Asynchronous Receiver Transmitter) des IGEPv2 verwendet, welche auf den RS-232-Standard konfiguriert wurde. Die Datenleitungen wurden mit dem zweiten und dritten Pin der J960-Verbindungsstelle (UART3) auf dem IGEPv2 verbunden und die UART-Konfiguration wie vom Hersteller vorgeschrieben⁹ umgesetzt. Die Stromversorgung des Pegelwandlers und somit auch der IMU erfolgte durch zwei

⁹http://labs.isee.biz/index.php/How_to_use_UARTs

Pins der J940-Verbindungsstelle, welche die erforderlichen Spannungen zur Verfügung stellen. Die Anbindung der Kamera erfolgte, wie dargestellt, per USB.

4.4 Kosten

Ein wichtiger Punkt bei der Auswahl der Hardware war neben technischen Aspekten auch der Kostenfaktor, welcher so gering wie möglich ausfallen sollte. Dieses hat den Zweck eine möglichst große Nutzergemeinde zu generieren und somit auch weniger finanzkräftigen Hochschulen einen Einstieg in diesen Forschungszweig zu bieten. Die Materialkosten des finalen Prototypen belaufen sich auf circa 500 Euro pro Einheit. Vergleicht man diese Summe mit einem häufig im RoboCup vertretenen Roboter, wie zum Beispiel dem „DARwIn-OP“¹⁰ von ROBOTIS, dessen Kosten sich ohne Rabatt auf 12000 US-Dollar, also circa 8900 Euro pro Einheit, belaufen, dann ist die finanzielle Ersparnis leicht ersichtlich.

¹⁰http://www.robotis.com/xe/darwin_ko

KAPITEL 5

Software

Nach der Konstruktion der Hardware musste die bestehende Software auf die neuen Gegebenheiten angepasst werden. Als Basis für die Entwicklung der Software wurde die FUmoids-Software genutzt. Die Software befand sich auf dem Stand der RoboCup-Weltmeisterschaft 2012 und wurde für diese Arbeit entsprechend angepasst. Der geschriebene Quellcode wurde dabei in die FUmoids-Software integriert, sodass die normale Funktionalität für die Roboter nicht beeinflusst wird. Für die beiden umzusetzenden Modi wurden, dem Framework (siehe Abschnitt 2.2) entsprechend, jeweilige Module und Repräsentationen implementiert.

5.1 Entwicklungsumgebung

Die Entwicklung findet unter Linux statt, wo mit Hilfe eines Cross-Compilers der Quellcode für ARM-Systeme übersetzt werden kann. Die so kompilierten Dateien werden mit Hilfe eines Installationsprogramms über das Netzwerk auf die gewünschten Prototypen kopiert. Viele zusätzliche Programme der FUmoids, ermöglichen eine Interaktion mit dem Hauptprogramm. So sind zwei Simulatoren vorhanden, in welchen eine virtuelle Umgebung eines Fußballspiels erzeugt wird, in der, der durch das Hauptprogramm erzeugte Roboter, mit der Umwelt interagieren kann und somit Softwaretests ohne realen Roboter möglich werden. Die Software „FUremote“ ermöglicht die Interaktion des Menschen mit dem Hauptprogramm, welche hauptsächlich zum Debuggen genutzt wird, aber auch zum Ändern der Roboterkonfiguration oder Kalibrierung dient. Zum Design des Verhaltens in XABSL wird ein eigener Editor bereitgestellt, in dem die Optionen, die Zustände und die Zustandsübergänge grafisch dargestellt werden, wodurch bei komplexen Verhaltensweisen eine verbesserte Übersicht möglich ist.

5.2 WLAN-Verbindung

Auf dem IGEPv2 dient ein Linux Kernel in Version 2.6.37 als Betriebssystem, welcher zusätzlich mit allen nötigen Programmen und Programmbibliothek erweitert wurde. In dieser Version des Betriebssystems wurde durch Messungen ein Problem entdeckt, welches die Nutzung der WLAN-Kommunikation verhinderte, da hohe Paketverluste keine stabile Kommunikation zuließen. In einer früheren Version (2.6.35) des zusammengestellten Betriebssystems trat das genannte Problem nicht auf, wobei eine nachträgliche Versionsänderung aufgrund fehlender Weiterentwicklung und Zeitgründen nicht umgesetzt wurde. Das Problem konnte nicht in einem adäquaten Zeitrahmen auf eine Ursache eingegrenzt werden, wodurch nur eine Kommunikation per Ethernet möglich war. Eine Fernkonfiguration der Prototypen und somit auch eine Kalibrierung auf den jeweiligen Menschen war somit während des RoboCups ausgeschlossen. Eine Kommunikation der Prototypen untereinander sowie mit dem Schiedsrichter-Programm war durch diesen Umstand ebenfalls nicht möglich, was die Möglichkeiten im Verhaltensdesign zusätzlich einschränkte. Die Protokollierung der internen Programmzustände und die Analyse dieser Daten zur Eingrenzung von beobachteten Problemen ist nur über das Netzwerk möglich, wofür während der Entwicklungsphase das Ethernet genutzt wurde. Probleme während der Demonstrationen konnten nur durch Beobachtung eingegrenzt werden und nachfolgend mit aktiver Netzwerkverbindung nachgestellt werden.

5.3 Audio-Framework

Um eine Verwendung durch andere Entwickler zu erleichtern, wurde beschlossen alle OpenAL-spezifischen Befehle zu abstrahieren, sodass für eine Nutzung der Audioausgabe keine OpenAL-Kenntnisse vonnöten sind. Ein weiterer Vorteil der Abstraktion ist ein möglicher Wechsel der Audio-API, ohne größere Code-Änderungen durchführen zu müssen. Kongruent zur bestehenden Struktur wurde ein Service implementiert, welcher jeder C++-Klasse Basisoperationen zur Beeinflussung der Audioausgabe bereitstellt. Der Service initialisiert die Audio-Hardware beim Programmstart, wie in den Grundlagen (siehe Abschnitt 2.6) beschrieben, und stellt seine Funktionen per „`services.getAudio()`“ zur Verfügung.

Der Aufbau des Service (siehe Abbildung 5.1) hat sich an der Funktionsweise von OpenAL orientiert, wodurch C++-Objekte für Audioquellen (`AudioSource`) und Audiopuffer (`AudioBuffer`) erstellt wurden, welche jeweils ihre spezifischen Funktionen zur Verfügung stellen. Diese Objekte werden durch Zeichenketten (Strings) identifiziert, was eine flexible Nutzung sicherstellen soll, da bei einer künstlich beschränkten Anzahl von Audio-Objekten zwar eine performantere Ansteuerung durch Zahlen oder Enums erfolgen könnte, jedoch eine erhöhte Kollisionswahrscheinlichkeit und einen erhöhten Verwaltungsaufwand mit sich bringen würde. Es müsste über mehrere Pro-

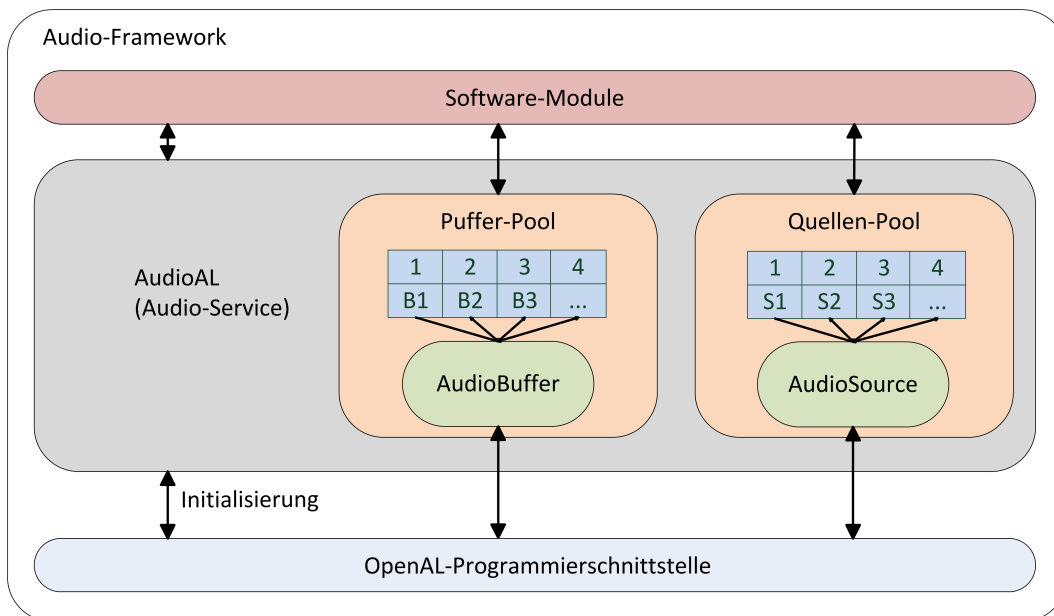


Abbildung 5.1: Visualisierung des Audio-Frameworks

grammteile und mehrere Entwickler hinweg eine Doppelnutzung eines Audio-Objekts verhindert werden, was im vorgestelltem Rahmen der Fumanoids sehr aufwändig wäre. Die verwendeten Strings hingegen bieten einen ausreichenden Namensraum, wodurch die fälschliche Nutzung fremder Audio-Objekte unwahrscheinlicher wird.

Im folgenden Beispiel 5.1 werden zuerst zwei Audiopuffer erstellt und mit Audiodaten befüllt. Mit der erstellten Audioquelle „Intro“ werden in der letzten Zeile beide Puffer (2) verknüpft, abgespielt und gewartet bis beide Puffer abgespielt wurden (true).

```

1      services.getAudio().createBuffer("dasTor", services.getAudio
      ().soundDirAndFile("df/dasTor.wav"));
2      services.getAudio().createBuffer("soKlingt", services.
      getAudio().soundDirAndFile("df/soKlingt.wav"));
3
4      services.getAudio().createSource(10, "Intro");
5
6      services.getAudio().playBuffersByName("Intro", true, 2, "
      soKlingt", "dasTor");

```

Quellcode 5.1: Audio-Beispiel

Alle erforderlichen Funktionen für die Nutzung von OpenAL werden durch den Service, die Audioquellen und die Audiopuffer bereitgestellt. Die Einhaltung spezifischer OpenAL-Bedingungen sowie die grundlegende Fehlerbehandlung erfolgt durch die genannten Klassen. Alle im Abschnitt 2.6 beschriebenen Eigenschaften der OpenAL-

Objekte werden unterstützt und sind durch entsprechende Funktionen modifizierbar.

5.4 IMU - UM6

Durch den Wechsel der IMU musste eine neue Schnittstelle geschaffen werden, welche das vom UM6 vorgegebene Protokoll unterstützt. Dafür wurde die Klasse „GyroUM6“ implementiert, welche die Kommunikation mit dem UM6 in einem eigenen Thread abwickelt. Alle vom UM6 unterstützten Befehle können gesendet werden, sowie alle erzeugten Daten ausgelesen werden. Durch einen fehlenden Interrupt ist es nötig ankommende Daten schnellstmöglich auszulesen und zu verarbeiten, da ansonsten empfangene Daten überschrieben und somit der komplette Datensatz ungültig werden könnte. Dafür ist ein separater Thread nötig, um den Ablauf des restlichen Programms nicht zu beeinflussen. Die benötigte Endlosschleife zum Auslesen der Daten wurde durch eine Unterbrechung der Programmausführung (usleep) optimiert, da ansonsten eine hohe CPU-Last auftritt. Durch die konfigurierte Baudrate des UM6 und dessen einstellbaren Transmissionsintervalle kann die zukünftige Kommunikationspause abgeschätzt und dementsprechend gewartet werden. Die Messwerte für Pitch-, Roll- und Yaw-Winkel werden durch den UM6 in definierten Abständen übermittelt und in die entsprechenden Strukturen der FUMANOIDs-Software für die spätere Verwendung gespeichert. Im Zuge dieser Umstellung wurden vorhandene Fehler in der FUMANOIDs-Software behoben und eine extra Klasse für die Nutzung von Winkeln erstellt, welche zukünftige Fehler gleicher Art verhindern soll.

5.5 Vision

Die Vision (Bildverarbeitung und Objekterkennung) ist die wichtigste Sensorquelle des Projekts und musste neben der üblichen Kalibrierung für das neue Szenario angepasst werden. Die markantesten Änderungen gegenüber den humanoiden Robotern ergab sich durch den veränderten Betrachtungswinkel des Menschen, die Nutzung eines größeren Balls und das größere Spielfeld. Die verwendete Vision basiert auf der Implementierung von Naja von Schmude [22] und Lisa Dohrmann [23]. Durch die robusten Algorithmen waren nur geringe Modifikationen nötig und die gute Dokumentation, in Form der Abschlussarbeiten, erleichterte das Verständnis sowie das Auffinden der beteiligten Quellcodeabschnitte. Die Anpassung der Vision erfolgte primär im Labor der FUMANOIDs, jedoch waren auch während des RoboCups Änderungen nötig, da durch spezielle Gegebenheiten vor Ort die Erkennungsleistung der Vision negativ beeinflusst wurde. So wurde zum Beispiel die Spielfeldkontur während des RoboCups nur unzureichend erkannt oder Bälle im Brauntönen der Tische am Spielfeldrand erfasst.

5.6 Modellierung

Die Modellierung der Objekte basiert auf der Masterarbeit von Stefan Otte [24] und benutzt einen *Extended Kalman Filter* zur Berechnung der zukünftigen Objektpositionen. Dabei wird jede Vorhersage durch eine nicht-lineare Funktion approximiert. In einem Korrekturschritt werden die Werte, welche zur Berechnung genutzt wurden, durch die Abweichung zur tatsächlichen Position korrigiert. Die Vorhersage wurde auch mit Hilfe der Odometrie berechnet, welche durch die Änderung der Hardware nicht mehr messbar war. Zusätzlich geben bestimmte Rauschwerte die Variabilität der einzelnen Parameter wieder, welche jedoch nur mit entsprechender Erfahrung angepasst werden können.

Die Modelle wurden in Vergleichstests den entsprechenden Perzepten gegenübergestellt und die Genauigkeit beurteilt. Dabei stellte sich heraus, dass die nicht modifizierten Modelle schlechtere Ergebnisse lieferten als die zugehörigen Perzepte. Basierend auf diesen Erkenntnissen wurden die Objekt-Perzepte statt der Objekt-Modelle für die weitere Verarbeitung genutzt, da die Modelle keinen essentiellen Mehrwert für eine prototypische Umsetzung dieser Arbeit darstellten, da ein Fußballspiel auch ohne vorhandene Modelle möglich ist.

5.7 Direct Mode

Als Vorbild für diesen Modus kann man Fledermäuse betrachten, da diese häufig in der Dunkelheit lebenden Tiere ebenfalls eine Adaption an ihre Umgebung vollzogen haben. Das gleiche Ziel verfolgt auch dieser Modus. Was bei Fledermäusen das fehlende Tageslicht ist, ist bei diesem Modus das fehlende Sehvermögen, wobei beide die Abwesenheit der visuellen Wahrnehmung zur Folge haben. Die Fledermaus hat dieses spezielle Problem durch das Aussenden und das Empfangen von Ultraschall gelöst, wodurch sie auch in der Dunkelheit „sehen“ kann. Sie „sieht“ also durch die Ohren, welche den reflektierten Schall und dessen Laufzeiten zu einem „Bild“ im Gehirn werden lassen. Das Ziel dieses Modus ist es, eine ähnliche Leistung mit Hilfe heutiger Technik zu ermöglichen, wobei die bereits beschriebene Hardware-Plattform im abgegrenzten RoboCup-Szenario genutzt werden soll.

5.7.1 Blindenfussball

Bei der Umsetzung einer solchen Aufgabe stößt man früher oder später auf die Frage, ob Blinde Fußball spielen und falls ja, wie genau sie diese Leistung vollbringen. Das Ergebnis ist, dass es in Deutschland seit dem Jahr 2008 eine Blindenfußball-Bundesliga¹ gibt. Genau wie im RoboCup sind spezielle Regeln nötig, um auf die Bedürfnisse der Spieler zu reagieren. Unter anderem ist der Ball mit Rasseln bestückt,

¹<http://www.blindenfussball.de>

was einen stetigen, ortbaren Ton erzeugt, sobald der Ball in Bewegung versetzt wird. Ein Spieler mit einer reduzierten Sehfähigkeit muss, ähnlich wie in dieser Arbeit, eine Dunkelbrille tragen, um Chancengleichheit herzustellen. Ein solches Spiel bietet eine hohe Dynamik, wodurch spezielle Schutzkleidung nötig wird, wie im Bild 5.2 zu erkennen ist. Eine ähnliche Leistung mit dem Computer zu erbringen, dürfte die Forschung die nächsten Jahre beschäftigen.



Abbildung 5.2: Saisonfinale der Blindenfußball-Bundesliga des Jahres 2013²

5.7.2 Umsetzung

Im Direct Mode wurden unterschiedlichste Experimente durchgeführt, um die Ortung einzelner Objekte zu verbessern sowie den vermittelten Gesamteindruck zu verbessern. Dafür wurden verschiedenartige Audiodaten verwendet und mit den Mitteln von OpenAL zusätzlich manipuliert. Es wurde ein Modul, entsprechend dem Blackboard-Konzept, erstellt, welches die verfügbaren Objektmodelle als Quelle für deren auditive Repräsentation nutzte. Die Funktion des Moduls wird mit dem Programmparameter „-dm“ aktiviert. Wie viel und welche Spielobjekte repräsentiert werden müssen, um einen Umgang mit dem Ball und das Erreichen des Tors zu ermöglichen, wurde in mehreren Versuchen ermittelt.

Die FHumanoid-Software liefert x- und y-Koordinaten der erkannten Objekte in Zentimetern relativ zum Betrachter, wobei ein planares Spielfeld angenommen wird und die Kameraposition sowie die Kameraausrichtung für die Projektion der Objektkoordinaten genutzt wird. Diese Koordinaten sind sowohl für Perzepte sowie Modelle verfügbar, wobei die Wahl, durch die bereits genannten Gründe, auf das Erstere fiel. Der Mensch „sieht“ also nur die Objekte, die im aktuellen Bild vom Vision-Modul erkannt wurden und kann sich dadurch sein eigenes Modell von der Umwelt machen.

Der einfachste Ansatz, die Koordinaten der erkannten Objekte unverändert an OpenAL weiterzureichen, stellte sich für wichtige Situationen als ungeeignet heraus. Beim Test

²<http://www.blindenfussball.de/?id=400&cmd=14&r=0&p=17>

dieser Repräsentation wurden weiter entfernte Objekte von nahen Objekten akustisch verdeckt. In einer optimalen Schussposition, wo Spieler, Ball und Tor in einer Linie standen, war das weiter entfernte Tor hinter dem vielfach lauterem Ball kaum zu orten. Dieses lag am Entfernungsmodell von OpenAL, was standardmäßig eine starke Dämpfung weit entfernter Objekte vornahm. OpenAL bietet eine Vielzahl von Einstellungen und Entfernungsmodellen, mit dem dieses negative Verhalten behebbar gewesen wäre. Statt den korrekten Modus mit den optimalen Parametern zu ermitteln, wurde eine andere Lösung gefunden, welche die Repräsentation unabhängig vom gewählten Entfernungsmodell macht. Dabei bewegen sich alle Tonquellen auf der Oberfläche einer Kugel mit dem Hörer im Zentrum der Kugel, wodurch alle Tonquellen äquidistant zum Hörer sind. In allen Entfernungsmodellen werden gleiche Distanzen auf gleiche Dämpfungen abgebildet. Somit besitzen alle Tonquellen mit gleicher individueller Lautstärke, unabhängig von ihrer Position auf der Kugeloberfläche, eine äquivalente Lautstärke in Bezug auf den Hörer. Eine akustische Übertönung anderer Tonquellen kann mit dieser Methode nicht ungewollt erzeugt werden, unabhängig vom gewählten Entfernungsmodell. Die Wahloption des Entfernungsmodells bleibt somit offen für andere Verwendungszwecke, bei denen ein spezielles Entfernungsmodell benötigt wird.

Mit dieser Repräsentation konnten Objekte bereits gut geortet werden, wobei ein Fehler von weniger als 10 Grad nur schwer zu erreichen war. Daher wurde ein zusätzlicher, akustischer Indikator verwendet um die Ortungsgenauigkeit weiter zu steigern. Die Frequenzen von Objekten wurden linear zum euklidischen Abstand zum Bildmittelpunkt erhöht. So haben Objekte am Rand einen tieferen Ton, als ein Objekt welches sich im Bildmittelpunkt befindet. Da das Gehör des Menschen die Frequenzspitzen sehr gut wahrnehmen kann, wird es so möglich den Zeitpunkt der maximalen Frequenz genau zu ermitteln und somit die Position eines Objekts bis auf wenige Grad Abweichung zu bestimmen. Der Mensch ist somit in der Lage ein präzises Modell seiner Umwelt zu entwerfen. Eine graphische Darstellung findet sich nachfolgend (siehe Abbildung 5.3), wo die zunehmende Dunkelheit eine steigende Frequenz darstellen soll. Je dunkler der Punkt, desto höher ist die Frequenz der Objekte, welche in diesem Bereich erkannt wurden.

Um eine Verwirrung bei der Unterscheidung von Objekten zu vermeiden, wurde in der letzten Version dieses Modus nur der Ball und das gegnerische Tor repräsentiert, welche zum Zweck des Torschießens ausreichende Informationen darstellen. Eine Repräsentation anderer Spieler wurde aufgrund einer potenziellen Informationsüberflutung verworfen, da mit den zusätzlichen Objekten die eigentliche Spielhandlung erschwert wird. Bei der Repräsentation des Tors stehen mehrere Koordinatenpaare zur Verfügung, so kann der linke oder rechte Pfosten sowie der Tormittelpunkt dargestellt werden, was allerdings eine subjektive Entscheidung ist, da keine der Möglichkeiten einen Informationsvorteil erzeugt.

Die verwendeten Audiodaten für Objekte sind allesamt Bruchstücke von aufgenommenen Tönen, welche in einer Endlosschleife abgespielt werden. Die Erzeugung pri-



Abbildung 5.3: Je dunkler das Gebiet, desto höher die Ausgabefrequenz

mitiver künstlicher Töne (Rechteck-, Sinus- oder Sägezahnsignal) brachte keine Verbesserungen, da die fehlende Varianz unangenehme Höreindrücke zur Folge hatte. Die natürlichen Tonbruchstücke hingegen besitzen genügend Dynamik, um während eines längeren Zeitraums subjektiv nicht als störend empfunden zu werden.

In der letzten Version dieses Modus gelang es circa einem Dutzend Personen einen erfolgreichen Torschuss, innerhalb von ein bis zwei Minuten nach dem Anlegen des Prototypen, zu erreichen.

5.8 Command Mode

Für den Command Mode wurde entsprechend dem Blackboard-Konzept ein Modul und eine Repräsentation erstellt, wobei Letztere hauptsächlich eine Schnittstelle zu XABSL darstellt. Die Funktion des Moduls wird mit dem Programmparameter „--cm“ aktiviert und der XABSL-Agent für das Verhalten wird mit „-behavior.agent==cchs“ konfiguriert. Das Verhalten, was die Menschen umsetzen sollen, ist in XABSL verfasst und lässt sich somit auf der höchsten Abstraktionsebene des Programms gestalten. Hierfür wurden unterschiedliche Gruppen von XABSL-Symbolen eingeführt, welche für die Interaktion mit dem Menschen benötigt werden und nachfolgend erklärt sind.

5.8.1 CCHS-Symbole

Befehlsausgabe

Um dem Träger des Prototypen Befehle geben zu können, wurden die XABSL-Symbole „cchs.word“, „cchs.end“ und „cchs.stop“ angelegt. Ein Satz wird wie bei natürlicher Sprache durch eine Kombination einzelner Wörter gebildet und mit einem Punkt beendet. Das Setzen von „cchs.word“ legt ein Wort aus dem Wortpool

in einer Warteschlange (Queue) ab. Mehrmaliges Setzen vergrößert die Warteschlange und verlängert somit den zu sprechenden Satz. Wurde der gewünschte Satz aus einzelnen Worten zusammengestellt, kann dieser durch das Setzen von „cchs.end“ abgeschlossen und anschließend automatisch abgespielt werden. Während ein Satz abgespielt wird, kann die Warteschlange nicht verändert werden. Ob eine Modifikation der Warteschlange aktuell möglich ist, kann durch das Lesen von „cchs.end“ festgestellt werden. Wird gerade ein Satz abgespielt liefert die Abfrage ein false zurück, wohingegen ein true die Modifizierbarkeit des Satzes signalisiert. Bedingt durch Änderungen der Umwelt kann es nötig sein, den aktuellen Befehl abzubrechen und einen zeitkritischen Kommando zu geben. Für diesen Fall beendet „cchs.stop“ die aktuelle Ausgabe, leert die Warteschlange und ist für neue Wörter bereit. Alle zur Verfügung stehenden Befehle wurden eigenständig aufgenommen und nachträglich bearbeitet, um eine möglichst klare Wiedergabe zu erreichen.

Folgendes XABSL-Beispiel zeigt in der ersten Codezeile das Erzeugen und Abspielen des Befehls „look down“. In der zweiten Zeile wird der vorherige Befehl abgebrochen, was einen sofortigen Schussbefehl ermöglicht.

```

1 action {cchs.word = LOOK; cchs.word = DOWN; cchs.end = true;}
2 ...
3 action {cchs.stop = true; cchs.word = SHOOT; cchs.end = true;}

```

Quellcode 5.2: XABSL-Befehlsbeispiel

Richtungsangaben

Das Symbol „cchs.watch“ dient zur akustischen Unterstützung von Befehlen, welche Richtungsangaben enthalten. Wird das Symbol gesetzt, kann ein Ton aus neun definierten Richtungen erzeugt werden. Die Richtungen bilden eine 3x3 Matrix, welche horizontal aus „LEFT“, „MIDDLE“ und „RIGHT“ besteht und vertikal aus „DOWN“, „MIDDLE“ und „UP“ besteht. Zum Abschalten des Tons wird das Symbol auf „OFF“ gestellt. Durch diese Symbole kann die Reaktionszeit den Menschen beschleunigt werden oder aber, bei entsprechender Einweisung, auf längere Befehle verzichtet werden. Eine Richtungsangabe kann dann ausschließlich über Töne erfolgen, was eine beschleunigte Befehlsumsetzung zur Folge haben könnte. Der abgespielte Ton ist hierbei an keine externe Quelle gebunden und somit nicht von der Kopfposition abhängig. Hingegen wird durch das boolesche Symbol „cchs.watch_play“ ein Ton erzeugt, welcher an bestimmte Kopfwinkel gebunden ist und sich dabei dynamisch an die Kopfposition anpasst. Auf die Kopfwinkel wird im folgendem Abschnitt 5.8.1 näher eingegangen, welche für die genannte auditive Repräsentation zwingend definiert sein müssen.

Im folgenden Beispiel wird ein Ton in der unteren linken Ecke erzeugt und im späteren Verlauf wieder deaktiviert, was eine Rotation nach links oder aber eine Blickanweisung darstellen kann.

```

1  action { cchs.watch = LEFT_DOWN; }
2  ...
3  action { cchs.watch = OFF; }

```

Quellcode 5.3: XABSL-Richtungsbeispiel

Feedback

Neben den Kamerabildern, welche von der Erkennungsleitung der Vision-Module abhängen, stellt die IMU die einzige Feedback-Quelle für das Verhalten dar. Daher wurde eine Struktur von XABSL-Symbolen erzeugt, welche Soll- und Ist-Zustände für jede Achse des Kopfes widerspiegelt. Damit wird es möglich eine bestimmte Position zu fordern und die Umsetzung der Forderung zu kontrollieren. Es wird ebenfalls die Zeit der Befehlsdurchführung gemessen, sodass verschiedene Eskalationsmöglichkeiten zeitgesteuert umgesetzt werden können. Mit „cchs.pitch_time“ kann die Zeit, welche seit dem Setzen von Winkel und Abweichung vergangen ist, in Millisekunden abgefragt werden. Für eine Auswertung des Ist-Zustandes kann „cchs.pitch_diff“ genutzt werden, welche die Differenz zum Soll-Zustand angibt. Je nachdem, ob die Differenz positiv oder negativ ist, kann auf die Richtung geschlossen werden, in welcher der Soll-Zustand liegt. Das boolesche Symbol „cchs.pitch_reached“ wird Wahr, sobald die Winkelbedingungen erfüllt sind und wieder auf Falsch gesetzt, wenn der geforderte Winkelbereich verlassen wird. Mit diesen Mitteln werden viele Möglichkeiten eröffnet, um im Verhalten auf die Kopfposition des Menschen auf unterschiedlichste Arten zu reagieren und entsprechende Befehle zu geben. Das Symbol „cchs.watch_play“ erzeugt an den definierten Winkeln einen Ton, wodurch eine Richtungsangabe in XABSL erleichtert wird.

Im folgenden Beispiel 5.4 wird ausgewertet, ob der Pitch- und Yaw-Winkel erreicht wurden und entsprechend reagiert. Im Action-Teil werden die geforderten Winkel und die zulässige Abweichung gesetzt, welche die notwendigen Werte für eine Berechnung darstellen.

```

1  initial state start {
2      decision {
3          if (cchs.yaw_reached && cchs.pitch_reached) goto reached;
4          else goto play;
5      }
6      action{
7          cchs.yaw_target = @yaw_target;
8          cchs.yaw_threshold = @yaw_threshold;
9          cchs.pitch_target = @pitch_target;
10         cchs.pitch_threshold = @pitch_threshold;
11     }
12 }

```

Quellcode 5.4: XABSL-Feedbackbeispiel

5.8.2 Verhalten auf dem RoboCup 2013

Während der Demonstrationen auf dem RoboCup stand nur die Befehlsausgabe in XABSL zur Verfügung. Durch diese und weitere Einschränkungen wurde nur ein einfaches Verhalten umgesetzt. Zuerst wurde der Ball durch ein bestimmtes Suchverhalten detektiert und anschließend zum Ball gegangen. Sollte der Ball währenddessen verloren gegangen sein, wurde eine erneute Ballsuche eingeleitet. Bei einer erfolgreichen Annäherung an den Ball, welche durch die Ballentfernung bestimmt wurde, wurde ein Schuss angewiesen. Eine Differenzierung der Spieler in unterschiedliche Rollen, wie Torwart oder Stürmer, fand nicht statt, wodurch alle Prototypen identische Verhaltensweisen besaßen.

Die Steuerung des Kopfes stellte die größte Schwierigkeit dar, da das Verhalten zu diesem Zeitpunkt keine Informationen über die Kopfposition besaß. Eine Reaktion auf falsch ausgeführte Befehle oder die Festlegung bestimmter Bewegungsziele war somit nicht möglich. Die Auswirkungen dieser Beschränkung werden in folgendem Abschnitt 5.8.3 näher erläutert.

5.8.3 Demonstration auf dem RoboCup 2013

Während des RoboCups 2013 wurden drei Demonstrationen durchgeführt, was die Aufmerksamkeit der Robocup-Teilnehmer sowie der Zuschauer weckte. Eine Demonstration beinhaltete mehrere Mini-Spielhälften von circa 5 bis 10 Minuten, zwischen denen die Prototypen an andere Interessenten weitergegeben wurden. Es gab auch Anfragen von Zuschauern, die gerne am Spiel teilnehmen wollten, was natürlich erfüllt wurde. Durch das rege Interesse wurden pro Demonstration bis zu fünf Spielhälften durchgeführt.



Abbildung 5.4: Neuausrichtung der Spieler nach einem Torschuss

Die Demonstrationen fanden immer spontan auf freien Spielfeldern der Teen-/Adult-Liga statt, was keine optimale Kalibrierung der Hardware auf die Gegebenheiten ermöglichte. Für eine präzise Projektion der Bildkoordinaten in Weltkoordinaten hätte auch eine genaue Erfassung und Konfiguration der jeweiligen Personengröße stattfinden müssen, was aufgrund zeitlicher Beschränkungen nicht möglich war. Diese und weitere Gründe, wie das jeweilige Verhalten der Menschen, waren für eher seltene Ballkontakte verantwortlich, wobei eine Steigerung von Demonstration zu Demonstration feststellbar war, da die gegebenen Befehle jeweils verbessert wurden.

Um einen korrekten Spielverlauf sicherzustellen und Verletzungen der teilnehmenden Personen zu verhindern, wurden, ähnlich wie bei den Roboter-Ligen die „robot handler“ besitzen, freiwillige „human handler“ genutzt. Dass eine solche Kontrolle nötig war, zeigte sich nicht zuletzt durch einige kleinere Zusammenstöße. Die Zusammenstöße der Körper der Menschen waren harmlos, da es nur zu geringen Fortbewegungsgeschwindigkeiten kam. Die Unsicherheit der Menschen, welche durch das Fehlen des Sehsinns erzeugt wird, führt somit automatisch zu einer erhöhten Aufmerksamkeit der restlichen Sinne und zu vorsichtigeren Bewegungen. Es kam allerdings auch zu Zusammenstößen zwischen den Prototypen, was durch die ungepolsterte Auflage auf der Nase schmerzhaft sein kann. In solchen Fällen sollten die „human handler“ rechtzeitig eingreifen, um Verletzungen zu vermeiden.



Abbildung 5.5: Bildung einer Gruppe am Ball

Das primitive Verhalten „Gehe zum Ball und schieße.“ führte öfters zu Gruppenbildungen, wie in Bild 5.5 zu erkennen. In einer solchen Gruppe ist es für die Vision schwer gewesen den Ball zu erkennen, da viele Annahmen der aktuellen Vision in einer solchen Situation nicht mehr zutreffen. Zum Beispiel wurde die Kontur des Spielfeldes durch größere Verdeckungen nicht mehr korrekt erkannt. Eine solche Verdeckung ist in Bild 5.6 zu erkennen, wo der Ball nur noch wenig erkennbar ist und

ein Großteil des Spielfeldes aus einer möglichen Spielerposition durch andere Spieler verdeckt wird.



Abbildung 5.6: Verdeckung des Balls durch Mitspieler auf dem Feld

Manche Personen agierten, sobald sie den Ball berührten, von alleine, was man sehr gut erkennen konnte. Der Ball wurde festgehalten und dribbelnd fortbewegt, was im Verhalten jedoch nicht vorgegeben wurde. So kamen unter anderem Hochschüsse zustande, was aufgrund fehlender Schutzreflexe bei den Mitspielern nicht unbedingt erwünscht ist. Inwieweit ein solch eigenmächtiges Verhalten eingeschränkt oder sogar geahndet werden sollte, muss bei der Definition der Spielregeln entschieden werden.

Die Suche des Balls erfolgte durch die Nachahmung eines Rechtecksignals, bei dem abwechselnd auf seine eigenen Füße („Look down.“) und wieder nach oben („Look up.“) geschaut werden sollte und anschließend eine Rotation nach rechts erfolgen sollte. Die Blickkommandos waren jedoch nicht eindeutig genug, was man sehr gut beobachten konnte. Bei den rotierenden Menschen auf dem Spielfeld konnten sehr gut die Extrema der Interpretationen beobachtet werden. Manche Personen drehten sich beim Rotationskommando um circa 90 Grad und wieder andere um weniger als 10 Grad, beides ungeeignete Methoden um den Ball auf dem Spielfeld effizient zu finden. Auch die vertikale Bewegung des Kopfes wurde ähnlich umgesetzt, wobei hier sogar Personen den Kopf überhaupt nicht senkten und immer geradeaus schauten.

Wurde der Ball erkannt, kam ein anderes Verhalten zum Einsatz, welches die vertikale Kopfausrichtung in drei Bereiche unterteilte, um den Ball nicht zu verlieren. Jedoch zeigte sich auch hier, dass die letzte Stufe „Look feet.“ ein häufiger Grund für den Ballverlust darstellte. Das Schauen auf die Füße war nötig, um den Ball im Nahbereich im Kamerabild zu halten, jedoch kam es wohl auch durch die zeitliche Verzögerung der Befehle immer wieder zu der Situation, dass bei Annäherung an den Ball dieser das Kamerabild nach unten hin verließ und in Folge der Ball erneut gesucht werden musste.

Durch diese Erkenntnisse bei der Kopfsteuerung des Menschen wurde nachträglich eine zusätzliche Möglichkeit geschaffen diese zu definieren und zu kontrollieren. Im

Verhalten kann nun ein genauer Winkel mit gewünschter Abweichung definiert werden, wodurch die genannten Missverständnisse ausgeräumt werden können. Die genaue Funktion ist im Abschnitt 5.8.1 beschrieben.

Es gab bei den Gesprächen mit den Nutzern des Prototyps auch hilfreiche Hinweise wie die Befehle effizienter übermittelt werden könnten. So gab der Prototyp bei der Demonstration alle zwei Sekunden einen Befehl, welcher je nach Befehlsdauer diesen Zeitraum besser (langer Befehl, kurze Pause) oder schlechter (kurzer Befehl, lange Pause) ausnutzte. Zum Beispiel wurde eine viel höhere Befehlsfrequenz von mehreren Befehlen pro Sekunde gefordert, wodurch schnellere Reaktionen auf die gegebenen Situationen möglich wären. Durch die Unterscheidung von Kopf- und Körperbewegungen waren längere Befehle zu Differenzierung nötig, was eine Umsetzung dieser Idee während des RoboCups verhinderte. Eine erneute Aufnahme aller Befehle, in einer schneller gesprochenen Version, zur Reduktion der Befehlsdauer, war aus Zeitgründen ebenfalls nicht möglich. Mit den später eingeführten, zusätzlichen Interaktionsmöglichkeiten, wie Richtungsangaben durch Töne, ist eine Differenzierung zwischen Kopf- und Körperbewegungen nicht mehr nötig. Die wegfallende Differenzierung ermöglicht eine Steigerung der Befehlsfrequenz, da die Befehle nun kürzer ausfallen.

Ein Vorschlag kam mit Anlehnung an die Befehlsform der Bundeswehr, dass die kommenden Richtungsangaben in den Befehl kodiert werden. So kann mit „Die Augen, links!“ oder „Augen, rechts!“ bereits mit dem ersten Wort die enthaltene Richtungsangabe ermittelt werden, was allerdings in dieser Verwendung zur Synchronisation der Bewegung dient. Unabhängig vom möglichen Nutzen wird dadurch jedoch eine Einweisung in die Befehle nötig und eine Eingewöhnungsphase, um die Umsetzung dieser zu trainieren, was im vorgestellten Rahmen nicht möglich gewesen wäre. Durch die bereits erwähnten Verbesserungen ist eine solche Kodierung nicht nötig, was den Vorteil einer wissensunabhängigen Nutzung des Prototypen mit sich bringt.

5.8.4 Regeln

In diesem Abschnitt sollen die Regeln einer möglichen, neuen Liga näher beleuchtet werden. Als Regelgrundlage hierfür dienen die offiziellen RoboCup-Regeln der Humanoid-Liga „RoboCup Soccer Humanoid League Rules and Setup“ [3]. Folgend werden alle essenziellen Regeländerungen erläutert, welche für die Anpassung an die Liga nötig erscheinen.

Spielfeld

Auf dem RoboCup 2013 wurden mehrere Demonstration mit Menschen auf dem Spielfeld der Adult-Liga, mit den Abmaßen 9 x 6 Meter, abgehalten. Bei den Spielen mit jeweils drei Mitspielern pro Mannschaft konnte festgestellt werden, dass es

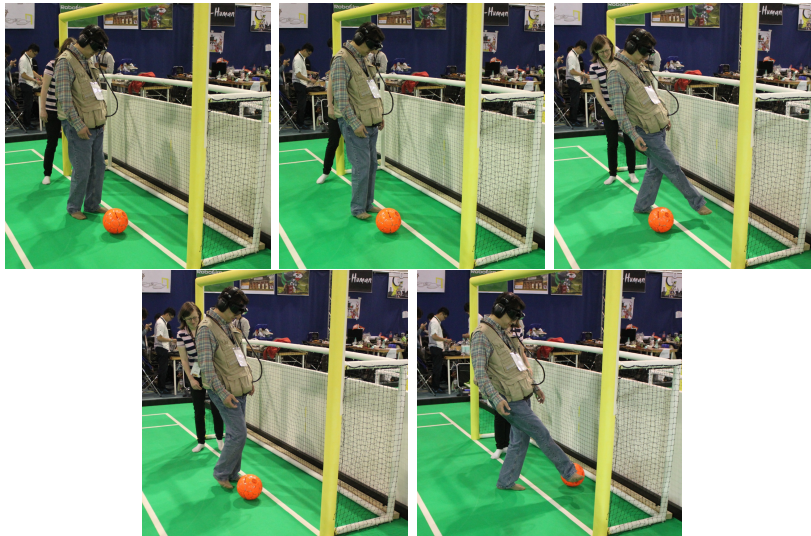


Abbildung 5.7: Sequenz von Bildern mit Torschuss

häufig zu Verdeckungen des Balls durch Mitspieler oder Gegner gekommen ist. Daher sollte bei der Umsetzung einer neuen Liga die Feldgröße angehoben werden. Die CCHS-Spiele könnten auf dem Spielfeld der Middle-Size-Liga [25] stattfinden, welche mit 18 x 12 Metern, eine vierfache Fläche zur Verfügung stellen würde und nach aktuellen RoboCup-Regeln das größte Spielfeld bietet. Auf einem solchem Spielfeld könnte die Spieleranzahl leicht angehoben werden. Nimmt man ein Hallenfußballfeld als Maßstab ³, welche mit 40 x 20 Metern fast die vierfache Größe eines Middle-Size-Liga-Spielfeldes bietet und auf dem sich maximal sechs Spieler pro Team befinden, dann sind maximal fünf Spieler pro Team auf dem Middle-Size-Liga-Spielfeld ein vernünftiger Kompromiss, da von einem ruhigerem Spielverlauf auszugehen ist. Sollte die Liga jedoch auf einem Adult-Size-Spielfeld stattfinden, dann ist die Spieleranzahl, aufgrund der eben genannten Größenverhältnisse, auf drei Personen pro Team zu begrenzen. Durch Weiterentwicklungen im Verhalten sollte es bei der gleichen Spieleranzahl, wie bei den diesjährigen Demonstrationen, zu einem strukturierterem Spielverlauf kommen, was zu weniger Verdeckungen markanter Objekte führt.

Ball

Bei der Demonstration konnten auch Schwächen in der Ballerkennung ausgemacht werden, welche an dem orangefarbenen Ball, mit dem üblicherweise bei den Humanoiden-Ligen gespielt wird, lagen. Der Farbton des Balls und der Hautfarbton der Spieler war mit der verwendeten Vision nur schwer zu differenzieren. Da die komplette Vermummung der Spieler als nicht zumutbar erscheint, sollte eine Änderung der Ballfarbe in Betracht gezogen werden. Hierfür bietet sich die Farbe Blau an, welche seit kurzem

³<http://www.dfb.de/index.php?id=11104>

nicht mehr als Ball-/Tor- oder Seitenpol-Farbe genutzt wird. Die Farbe Blau in der Bekleidung der Spieler zu vermeiden, ist mit minimalem Aufwand zu erreichen und orientiert sich an bereits bestehenden Bekleidungsvorschriften für Schiedsrichter und Robot-Handler.

Da viele Schutzreflexe des Menschen auf dem Sehsinn basieren, welche in diesem Szenario nicht mehr wirksam sind, sollte zum Schutz der Mitspieler auf eine weiche Ballalternative, wie einen Softball, zurückgegriffen werden, wodurch das Verletzungsrisiko bei Hochschüssen sinkt.

Spieler

Wie im vorherigen Abschnitt bereits beschrieben, sollten Spieler eine vorgeschriebene Kleidung tragen, welche nicht mit den anderen Spielfarben kollidiert. Des Weiteren soll angemerkt werden, dass es sinnvoll sein kann, die Größe oder das Gewicht der Spieler einzuschränken. Der recht variable Prototyp, welcher einer größtmöglichen Personengruppe passen sollte, wurde bei den Demonstrationen bereits bis auf die maximalen Anpassungsparameter variiert, um allen Spielern zu passen. Auch bei Zusammenstößen zwischen Spielern, welche bei den Demonstrationen unverletzt vonstatten gingen, sollte eine ähnliche Statur der Spieler zur Reduktion des Verletzungsrisikos beitragen.

Sensoren

In der CCHS-Liga sollten dieselben Sensoren wie in der Humanoid-Liga verwendet werden, da sonst die gewünschte Entwicklungsbeschleunigung ausbleiben könnte. In dieser Regelübernahme ist bereits eine implizite Verschärfung der Regeln enthalten, da sämtliche Informationen der Motoren wegfallen, welche aktuell noch eine wichtige Datenquelle darstellen.

Eigenmächtiges Verhalten

Während der Demonstration nahmen auch Personen am Spiel teil, welche nicht nur als Akteur agierten, sondern mit den ihnen zur Verfügung stehenden Informationen selbstständige Aktionen ausführten. Ob ein solches Verhalten gewünscht oder missbilligt wird, muss im weiteren Verlauf geklärt werden. Bei einem Schwerpunkt auf medienwirksamen Spielen könnte ein solches Verhalten gebilligt werden, wobei damit kein direkter Fortschritt in der Forschung zu erwarten ist. Sollte der Schwerpunkt auf die Forschung gelegt werden, dann muss ein solches Verhalten rechtzeitig erkannt und mit den üblichen Methoden unterbunden werden.

Die Erkennung von eigenmächtigen Aktionen lässt sich durch unterschiedlichste Methoden umsetzen. So könnte ein definierter Befehlspool erstellt werden, welcher den Spielern nur bestimmte Aktionen zur Verfügung stellt und somit eine einheitliche

Überwachung ermöglicht. Durch eine Protokollierung der gegebenen Befehle könnte ein Verstoß im Nachhinein durch eine Videoaufzeichnung festgestellt werden, was jedoch sehr zeitintensiv und nicht spielbeeinflussend wäre. Die erhaltenen Kommandos und die Reaktion der Spieler könnte durch eine weitere Person noch während des Spiels beobachtet und beurteilt werden, was direkte Einflüsse auf das Spiel hätte, jedoch zusätzlichen Personalaufwand bedeutet.

Das Einsetzen unbeteiligter Personen als Spieler ist jedoch die einfachste Möglichkeit der Kontrolle. Diese könnten eventuelle Verstöße melden und würden eine einsteigerfreundliche Befehlsstruktur voraussetzen. Bei dieser Methode könnten Personen aus dem Publikum als Spieler agieren, wodurch zusätzliches Interesse generiert werden könnte.

Als letzter Punkt sei noch die Option einer Fußpolsterung genannt, welche den natürlichen Umgang des Menschen mit dem Ball erschweren würde. Die Formgebung könnte eine vereinfachte Modellierung des Fußes auf Softwareebene ermöglichen und die Polsterung unerwünschte Verhaltensweisen, wie das Festhalten oder das Erfühlen des Balls, erschweren.

KAPITEL 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

In der vorliegenden Arbeit wurde ein neues Konzept für die humanoide RoboCup-Liga vorgestellt und umgesetzt, welches zu einer Beschleunigung der Roboterentwicklung beitragen soll. Dabei dient ein Mensch als humanoide Plattform für die übliche Roboterhardware, wobei dieser durch akustische Informationen gesteuert wird. Im Rahmen dieser Arbeit wurden mehrere Hardware-Prototypen konstruiert und etwaige Schwächen in Nachfolgeversionen behoben. Für mehrere Demonstrationen während des RoboCups 2013 wurden acht identische Prototypen montiert und mit der entwickelten Software ausgestattet.

Die FUMANOIDs-Software aus dem Jahr 2012 diente als Basis für die Prototypen und wurde um die benötigten Komponenten erweitert. Eine Ansteuerung der Audio-Hardware, die Einbindung eines neuen Orientierungssensors sowie die prototypische Umsetzung der beiden Modi *Direct Mode* und *Command Mode* wurden implementiert. Die bestehenden Softwarekomponenten wurden zusätzlich für eine Nutzung am Menschen modifiziert.

Die Demonstrationen des Command Mode auf dem RoboCup wurden erfolgreich durchgeführt und stießen auf Interesse seitens der RoboCup-Teams sowie der Zuschauer. Das Verhalten der spielenden Probanden zeigte, abhängig von der jeweiligen Person, viele Parallelen zu humanoiden Robotern. Manche der Personen befolgten die erhaltenen Befehle exakt, was aufgrund des übersichtlichen Verhaltens gut zu beobachten war. Andere Personen jedoch analysierten die erhaltenen Befehle, um sich ein Modell ihrer Umwelt zu kreieren und darauf basierend eigene Aktionen durchzuführen.

Die sichtbaren Schwächen der Software wurden teils auf dem RoboCup, teils danach behoben. Für eine exakte Positionierung des Kopfes wurde nachträglich ein Feedback-Kanal für das Verhaltensdesign geschaffen, welcher auf dem Orientierungs-

sensor basiert.

Für den Direct Mode wurde eine auditive Repräsentation gefunden, welche es vielen Personen ermöglichte, nach einer kurzen Eingewöhnungsphase, Tore zu schießen.

6.2 Ausblick

6.2.1 Hardware

Wie bereits im Kapitel 2 erläutert, existieren mehrere Informationskanäle zum Menschen, welche durch den Ausbau der Hardware erschlossen werden können. Als Hardwareerweiterung sind viele Möglichkeiten denkbar, jedoch sollte immer der Kostenfaktor im Hinterkopf behalten werden. Nachfolgend sollen ein paar Anregungen genannt werden, welche zur Verbesserung der Plattform beitragen könnten.

taktiler Output Auch kann über einen taktilen Output nachgedacht werden. Vibrationsmotoren, wie sie im Handys vorkommen, sind eine günstige Möglichkeit einen weiteren Informationskanal zum Menschen zu öffnen. Über Positionierung des Motors, wie Beine oder Arme, sowie über die Vibrationsfrequenz könnten eine Vielzahl von eindeutigen Befehlen übermittelt werden. Der finanzlastigste Teil dieser Idee wird die Ansteuerung und Energieversorgung eines solchen Motors sein. Diese Idee liefert allerdings nur Verbesserungen in einem sekundären Bereich, der Mensch-Maschine-Interaktion, wodurch hier besonders auf das Kosten-Nutzen-Verhältnis geachtet werden sollte.

Videobrille Der Einsatz einer Videobrille ist nur im „Command Mode“ sinnvoll, würde diesen aber um den primären menschlichen Wahrnehmungssinn erweitern. Auf dem Display könnten mehrere Symbole oder Animationen eindeutige Befehle vermitteln, welche im Gegensatz zum akustischen Informationskanal eine wesentlich höhere Bandbreite aufweisen. Bei der Entwicklung von neuen Vision-Algorithmen könnte das Ergebnis ohne Verzögerungen direkt auf dem Spielfeld betrachtet werden. Die Informationsübermittlung an den Menschen würde erleichtert, wodurch weniger Fokus auf eine geeignete auditive Repräsentation gesetzt werden müsste. Allerdings hat diese Erweiterung der Hardware einen entscheidenden finanziellen Nachteil, da sich die Kosten einer Einheit, nach heutigen Preisen, mindestens verdoppeln würde. Ungeachtet des Preises würde diese Erweiterung einen hohen Mehrwert für die Plattform bedeuten.

Kameramotoren Bei der aktuellen Plattform muss im „Command Mode“ ein Teil des Informationskanals genutzt werden um den Blick den Menschen in eine gewünschte Richtung zu lenken. Eine Verbesserung wäre durch jeweils einen Pitch- und einen Yaw-Motor zu erreichen, welche auch bei aktuellen Robotern für den Zweck der Kamerapositionierung genutzt werden. Die Software hätte somit vom Menschen unabhängige Aktuatoren um die Kameraausrichtung zu steuern. Nachteile dieser Verbesserung wären ein erhöhter Energiebedarf sowie

gesteigerte Kosten in Höhe von zwei Motoren, was die bisher teuerste Änderung darstellen dürfte.

Mehrkernprozessor Da die aktuelle Plattform nur einen Einkernprozessor verwendet, ist die Rechenleistung limitiert. Es könnte auf einen modernen Mehrkernprozessor gewechselt werden, welcher genug Rechenleistung für komplexere Berechnungen zur Verfügung stellt. Dieses würde zum Beispiel ermöglichen, die Kameraauflösung zu erhöhen, wodurch die Objekterkennung auf einem detaillierteren Bild stattfinden würde. Die Kosten einer solchen Plattform sind dabei ähnlich zum verwendeten IGEPv2, weshalb nichts gegen diese Verbesserung in einer späteren Version spricht.

GPS Da das langfristige Ziel des RoboCups ein Fußballspiel auf einem normalem Spielfeld ist, würde ein zusätzliches GPS-Modul eine einfache Lokalisierung auf diesem ermöglichen. Die neuen Informationen könnten eine visuelle Lokalisation ersetzen oder aber in eben diese Berechnung mit einfließen.

6.2.2 Software

Das größte Potenzial steckt jedoch in der Verbesserung der Software, da diese, entsprechend der Grundidee, nun unabhängiger von der Hardware weiterentwickelt werden kann. Durch die Modellierung von Objekten und die Lokalisation des Menschen stehen für das Verhaltensdesign wesentlich mehr Informationen zur Verfügung, welche ein verbessertes Spielverhalten ermöglichen würden, daher sollten diese Software-Module mit hoher Priorität für diese Plattform angepasst werden.

Die Vision benötigt in Zukunft eine Anpassung an die noch zu definierenden Spielparameter, wie die Spielfeldgröße, die Spielerfarben oder die Ballfarbe. Nach einer Definition dieser Parameter kann die Erkennung entsprechender Objekte implementiert werden, wodurch zum Beispiel der Mensch in der Lage wäre, unnötige Kollisionen mit Mitspielern zu vermeiden. Angepasste Objektmodelle würden eine Berücksichtigung von Objekten außerhalb des Kamerabildes ermöglichen, was im Falle des Direct Mode eine verbesserte Repräsentation der Umwelt ermöglicht.

Die aktuell genutzte Kameraauflösung ist für größere Spielfelder zu gering und bedarf einer Anpassung, da weit entfernte Bilddetails nur wenige Bildpunkte (Pixel) ausfüllen und schon durch das Rauschen des Sensors verloren gehen können. Die Spielfeldlinien (12 Zentimeter breit) am gegnerischen Tor sind schon auf einem Kid-Size-Feld über die volle Spielfeldlänge (9 Meter) kaum zu erkennen, da diese bei einer vertikalen Auflösung von 480 Pixeln nur wenige Pixel umfassen. Da die Spielfeldlinien jedoch wichtige Orientierungspunkte darstellen, muss mit wachsender Spielfeldgröße auch die Kameraauflösung erhöht werden, da sonst wichtige Informationen verloren gehen. Eine gesteigerte Auflösung vervielfacht die zu analysierenden Pixel im Bild, was wiederum zu einem erhöhten Bedarf an Rechenleistung führt. Da die Rechenleistung ein begrenzter Faktor ist und nicht beliebig erhöht werden kann, verlangt es in Zukunft nach effizienteren Algorithmen für die Bildanalyse.

Die mobile Plattform ermöglicht zudem eine Nutzung unter freiem Himmel, was aufgrund stark variierender Umweltbedingungen viele neue Herausforderungen für die Bildverarbeitung und Objekterkennung mit sich bringt. Für das Ziel von Roboter-Weltmeistern im Jahre 2050 ist es jedoch unerlässlich diesen Schnitt in möglichst naher Zukunft zu tun.

In der Art der Repräsentation des Direct Mode liegt noch viel Optimierungspotenzial, so könnten speziell erzeugte Töne eine gewünschte Ortbarkeit besitzen, sodass zum Beispiel ein näher kommender Ball mit sinkender Entfernung genauer ortbar wird oder aber ein näher kommendes Tor durch die Ortungsgenauigkeit eine virtuelle Torbreite vermittelt. Bei diesen und weiteren Optimierungen sollte nicht außer acht gelassen werden, dass die auditive Wahrnehmung eines jeden Menschen unterschiedlich ist und somit das Finden von generischen Lösungen erschwert wird. Zudem könnten Versuchsreihen durchgeführt werden, in denen Menschen unter bekannten Bedingungen bestimmte auditive Informationen erhalten, wodurch eine objektive Beurteilung der gewählten akustischen Repräsentationen möglich wird.

Der Command Mode könnte ebenfalls durch objektive Messungen optimiert werden, da eine geringe Reaktionszeit des Menschen bei der Umsetzung der Befehle zu einem verbessertem Spielfluss beiträgt. Ein Grundstein für solche Messungen wurde mit der Feedback-Funktion des Command Mode bereits implementiert, womit die Ausführungszeit einer bestimmten Kopfposition gemessen werden kann. Des Weiteren kann mit unterschiedlichsten Methoden die Befehlsübermittlung zum Menschen erforscht und optimiert werden. Auch eine Profilbildung wäre vorstellbar, in der die Befehlsübermittlung für eine bestimmte Person individuell abgestimmt wird. Die unterschiedlichen Informationskanäle oder Befehlsformen könnten in einer Evaluations- und Kalibrierungsphase, basierend auf Messergebnissen, variabel gewichtet werden und so eine optimale Schnittstelle zu einem bestimmten Menschen aufbauen.

Die Ermittlung der Kamerahöhe könnte durch eine automatische Kalibrierung erfolgen und somit genauere Projektionen ermöglichen. Diese ändert sich auch beim Bewegen des Kopfes, wie bei einem humanoiden Roboter, jedoch können diese Schwankungen für die festen Abmaße eines Roboters berechnet und korrigiert werden. Eine ähnliche Kompensation könnte für einen Menschen möglich werden, wenn dieser unterschiedliche Kopfpositionen zur Erzeugung eines entsprechenden Modells einnimmt und daraus eine dynamische Kamerahöhe berechnet wird. Man kann sich leicht vorstellen, dass die Kamera beim Geradeausschauen mehr als 10 Zentimeter höher ist, als wenn man direkt nach unten auf seine Füße schaut, wodurch sich in späteren Berechnungen unnötige Fehler ergeben.

Die Anpassung aller Objekt-Modelle an die neuen Rahmenbedingungen würde einen Mehrwert für das Projekt bedeuten und sollte bis zur nächsten Demonstration umgesetzt werden. Dadurch würden komplexere Verhaltensweisen möglich, welche einen positiven Effekt auf das Spielverhalten haben sollten. So könnte ein Torwart in Bedrängnis immer eine Position relativ zum eigenem Tor einnehmen, sodass ein gerader Schuss des Gegners blockiert wird, wofür jedoch ein akkurates Tormodell benötigt

wird. Ob eine sinnvolle Modellierung mit den gegebenen Daten möglich ist, ist in einer separaten Arbeit zu prüfen, da durch den Wegfall der Odometrie keine Daten über die Translation des Menschen mehr vorhanden sind. Das System kann also keine Fortbewegung auf dem Spielfeld feststellen, wodurch externe Informationsquellen wie eine Lokalisation hinzugezogen werden müssten, um eine Positionsänderung auf dem Spielfeld festzustellen.

Die fehlende Kommunikation über WLAN muss in naher Zukunft behoben werden, da ansonsten viele Funktionalitäten der Software nicht genutzt werden können. So ist ohne einen Kommunikationskanal zwischen den Spielern kein koordiniertes Spiel möglich und auch eine zentrale Steuerung des Spielverlaufs per Schiedsrichter-Programm entfällt. Daher sollte die Reaktivierung der WLAN-Kommunikation eine hohe Priorität besitzen.

6.2.3 RoboCup

Aufgrund der positiven Resonanz wird es hoffentlich auch im Jahr 2014 eine Demonstration von CCHS geben, welche letztendlich in einer neuen RoboCup-Liga enden könnte. Um dieses Ziel zu erreichen müssen feste Regeln etabliert werden, damit eine Umsetzung auf Softwareebene möglich wird. Ähnlich wie in jeder RoboCup-Liga ist ein Regelwerk nötig, welche die Umgebung des Spielers genauestens definiert, die erlaubte Hardware beschränkt und alle weiteren das Spiel beeinflussenden Faktoren regelt. Ein vorübergehendes Regelwerk sollte zum RoboCup 2014 vorhanden sein, damit die angestrebte Demonstration nach bekannten Maßstäben beurteilt werden kann.

Zudem könnte über eine alternative Nutzung der Plattform im Rahmen des RoboCup@Home nachgedacht werden. Ein Assistenzsystem für Blinde, was die Orientierung und Navigation in unbekanntem Umgebungen ermöglicht. Die Interaktion mit dem Menschen und die Erkennung von Objekten sind in dieser Liga eine Grundvoraussetzung. Statt einem Roboter Befehle zu geben, welcher diese für einen umsetzt, könnte die Plattform als ein interaktives Auge dienen, was die Realität eines Blinden um zusätzliche Informationen erweitert.

Abbildungsverzeichnis

1.1	Einleitung - Beispielroboter	1
1.2	Einleitung - NAO	3
1.3	Einleitung - Demonstration auf dem RoboCup 2013	4
1.4	Einleitung - Logo der FUmanoids	6
2.1	Grundlagen - Optische Illusion	7
2.2	Grundlagen - Hörfläche	10
2.3	Grundlagen - Achsen des Kopfes	11
2.4	Grundlagen - Cone of Confusion	12
2.5	Grundlagen - Modulabhängigkeiten	14
2.6	Grundlagen - Berlin United-Framework	15
2.7	Grundlagen - Roll-Pitch-Yaw-Winkel	16
2.8	Grundlagen - XABSL-Option	18
2.9	Grundlagen - Grundobjekte OpenAL	19
3.1	Ähnliche Arbeiten - Flugverkehr	22
3.2	Ähnliche Arbeiten - Erweiterte Realität	23
3.3	Ähnliche Arbeiten - Oculus Rift	24
4.1	Hardware - Prototyp 1	25
4.2	Hardware - Prototyp 2	26
4.3	Hardware - RoboCup-Prototyp	27
4.4	Hardware - Kamera	28
4.5	Hardware - IGEPv2	28
4.6	Hardware - Akkumulator	29
4.7	Hardware - Inertialsensor	30
4.8	Hardware - Anpassungsdauer der IMU	32
4.9	Hardware - Lanzzeitdrift der IMU	33
4.10	Hardware - Pegelwandler	33
4.11	Hardware - Arbeitsschutzbrille	34
4.12	Hardware - Gehörschutzkopfhörer	34
4.13	Hardware - Grundaufbau des RoboCup-Prototypen	35

5.1	Software - Audio-Framework	39
5.2	Software - Blindenfußball	42
5.3	Software - Frequenzänderung	44
5.4	RoboCup - Spielerausrichtung	47
5.5	RoboCup - Gruppenbildung	48
5.6	RoboCup - Verdeckungen	49
5.7	RoboCup - Schussesequenz	51

Literaturverzeichnis

- [1] Daniel Seifert and Stefan Otte and Johannes Kulick and Naja von Schmude and Lisa Dohrmann and Steffen Heinrich and Hamid Moballeggh and Sebastian Mielke and Lutz Freitag and Simon Hohberg and Julius Auer and Max Losch and Raúl Rojas. Berlin United - FHumanoids Team Description Paper 2012. In Xiaoping Chen and Peter Stone and Luis Enrique Sucar and Tijn Van der Zant, editor, *RoboCup 2012: Robot Soccer World Cup XV Preproceedings*. RoboCup Federation, January 2012.
- [2] *Plück-Rüsse*. Meyers neues Lexikon. 1975.
- [3] RoboCup Soccer Humanoid League Rules and Setup For the 2013 Competition in Eindhoven, May 2013.
- [4] Gegenfurtner, Karl R. *Gehirn und Wahrnehmung: eine Einführung*. Fischer-Taschenbuch-Verl., Frankfurt am Main, Aktualisierte Aufl. edition, 2011.
- [5] Effenberg, Alfred O. *Sonification: ein akustisches Informationskonzept zur menschlichen Bewegung*. Hofmann, Schorndorf, 1996.
- [6] Dürrer, Bernd. Untersuchungen zum Entwurf von Auditory Displays. Universitätschrift, Berlin, 2001.
- [7] FHumanoids. FHumanoids Code Release 2012, December 2012. Only available online: <http://www.fumanoids.de/publications/coderelease>.
- [8] Daniel Seifert and Raúl Rojas. FHumanoids Code Release 2012. In *In Proceedings of 17th RoboCup International Symposium*, June 2013. Eindhoven, Netherlands.
- [9] CH Robotics. Understanding Euler Angles, 2013. Revision 1.1.
- [10] Martin Löttsch and Joscha Bach and Hans-Dieter Burkhard and Matthias Jünger . Designing Agent Behavior with the Extensible Agent Behavior Specification Language XABSL . In Daniel Polani and Brett Browning and Andrea Bonarini , editor, *RoboCup 2003: Robot Soccer World Cup VII* , volume 3020 of *Lecture Notes in Artificial Intelligence* , pages 114–124 , Padova, Italy , 2004 . Springer .
- [11] Garin Hiebert. OpenAL 1.1 Specification and Reference, 2006. Revision 2.8.
- [12] Creative Technology Limited. OpenAL Programmer's Guide, 2007. Revision

- 1.5.
- [13] Joffrion, Jacque M. HEAD TRACKING FOR 3D AUDIO USING A GPS-AIDED MEMS IMU. Technical report, DTIC Document, 2005.
 - [14] Frauenberger, Christopher and Noisternig, Markus. 3D audio interfaces for the blind. In *Workshop on Nomadic Data Services and Mobility*, pages 11–12, 2003.
 - [15] Sundareswaran, Venkataraman and Wang, Kenneth and Chen, Steven and Behringer, Reinhold and McGee, Joshua and Tam, Clement and Zahorik, Pavel. 3D audio augmented reality: implementation and experiments. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, page 296. IEEE Computer Society, 2003.
 - [16] Raúl Rojas and Hamid Mobalegh and Björn Karger. Computer Controlled Human Soccer League: A new step towards the 2050s goal. June 2013.
 - [17] ISEE - Integration Software & Electronics Engineering. IGEP TM v2 Hardware Reference Manual, 2013. 2.0.
 - [18] reichelt elektronik GmbH & Co. KG. Mobil-Akku VTB-22 Bedienungsanleitung, 2012.
 - [19] CH Robotics LLC. UM6 Ultra-Miniature Orientation Sensor Datasheet, 2013. 2.3.
 - [20] SparkFun Electronics. RS232 Shifter SMD v1.0, 2005.
 - [21] UVEX ARBEITSSCHUTZ GmbH. uvex Schutzbrillen-Katalog 2013, 2013.
 - [22] Naja von Schmude. Farb- und kantenbasierte objekterkennung humanoider roboter im robocup-szenario. Diplomarbeit, Freie Universität Berlin, November 2011.
 - [23] Lisa Dohrmann. Kalibrierungsfreie Vision für humanoide Fußball-Roboter. Masterarbeit, Freie Universität Berlin, February 2012.
 - [24] Stefan Otte. Where am i? what's going on? – world modelling using multi-hypothesis kalman filters for humanoid soccer robots. Master thesis, Freie Universität Berlin, December 2012.
 - [25] Middle Size Robot League Rules and Regulations for 2013, January 2013.