



JOHANNES KEPLER
UNIVERSITÄT LINZ

Mathematik für Elektrotechnik, Informatik und Physik



TangibleControl – Universelle TUIs mittels Accelerometer

Diplomarbeit

zur Erlangung des Akademischen Grades

Diplom-Ingenieur

in der Studienrichtung

Informatik

Angefertigt am Institut für *Pervasive Computing*.

Betreuung:

Univ. Prof. Mag. Dr. Alois Ferscha

Eingereicht von:

Martin Reichör

Beurteilung:

Univ. Prof. Mag. Dr. Alois Ferscha

Linz, Oktober 2004

Kurzfassung

Es wurde ein Tangible User Interface (TUI) entwickelt welches es ermöglicht Gesten des Benutzers zu erkennen, abhängig davon können beliebige Aktionen ausgeführt werden. Einführend werden die Entstehung von Ubiquitous Computing und die Einordnung von TUIs in dieses Gebiet erläutert, gefolgt von der Bewegungserkennung mittels TUIs im Vergleich zu Computer Vision und der Erkennung mittels Datenhandschuhen.

Nach einem Überblick über die Technologien zur drahtlosen Datenübertragung und der daraus resultierenden Wahl von Bluetooth zur Übertragung der Sensordaten wird näher auf die entwickelte Hard- und Software eingegangen. Als Sensor wird ein Accelerometer mit drei Freiheitsgraden verwendet, hiermit wird die Lage im Raum bestimmt auf diesen Sensorwerten basiert die Gestenerkennung. Die Hardware kann mobil und autonom betrieben werden.

Neben mehreren Anwendungen zur Visualisierung der Sensorwerte wurde ein Mediaplayer entwickelt, hierbei können alle wesentlichen Funktionen durch Manipulation der TUIs, es handelt sich um eine Getränkedose und eine Zigarettenschachtel, ausgeführt werden.

Abstract

A Tangible User Interface (TUI) was developed which it makes possible to recognize gestures of the user, dependent on it arbitrary actions can be executed. By way of introduction the emergence by Ubiquitous Computing and the classification of TUIs into this area are described, followed of the gesture recognition by means of TUIs compared with computers vision and the recognition by dint of data gloves.

After an overview of the technologies to the wireless data communication and the choice from Bluetooth to the transmission of the sensor data, resulting from it, the hard- and software is described more detailed. A Accelerometer with three degrees of freedom is used as sensor, hereby the gesture recognition is based on the orientation of the sensor. The hardware can be operated mobilely and autonomously.

Apart from several applications for the visualization of the sensor values a Mediaplayer was developed. All substantial functions can be executed by manipulation of the TUIs, these are a beverage box and a cigarette box.

Danksagung

An dieser Stelle möchte ich all jenen danken, die durch ihre fachliche und persönliche Unterstützung zum Gelingen dieser Diplomarbeit beigetragen haben.

Prof. Ferscha der mir diese Arbeit ermöglichte und mich stets mit gutem Rat durch diese Diplomarbeit geleitet hat

Den Mitarbeitern des Institutes für Pervasive Computing für die freundliche Unterstützung.

Weiters möchte ich mich bei den Studienkollegen Harald, Rene und Thomas für die nicht nur fachliche Hilfe vor und während der Diplomarbeit bedanken.

Ein Dank gilt auch meinen Freunden, im speziellen Daniel, Evelyn, Michael, Martin und Reinhard.

Besonderer Dank gebührt meinen Eltern, die mir dieses Studium durch ihre Unterstützung ermöglicht haben, sowie meiner Familie.

Inhaltsverzeichnis

Kurzfassung	II
Abstract	III
Danksagung	IV
Inhaltsverzeichnis	V
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	IX
1 Einleitung	1
1.1 Motivation	1
1.2 Problemstellung	1
1.3 Überblick	4
2 Ubiquitous Computing	6
2.1 Entstehung von Ubiquitous Computing	6
2.2 Gründe, die Ubiquitous Computing ermöglichen	7
2.3 Einordnung der Virtualitäts- und Realitätsbegriffe	7
2.3.1 <i>Embodied Virtuality</i>	8
2.3.2 <i>Augmented Reality</i>	8
2.3.3 <i>Mixed Reality</i>	8
2.4 Tangible User Interface (TUI)	8
2.4.1 <i>Graspable User Interfaces</i>	8
2.4.2 <i>Von GUI zu TUI</i>	9
2.4.3 <i>Tangible Bits</i>	10
2.4.4 <i>Abgrenzung von TUI zu andern Konzepten</i>	12
2.5 Kontext	12
2.6 Multimodale Interaktion	14
2.7 Beispielhafte Anwendungen	15
2.7.1 <i>DigitalDesk</i>	15
2.7.2 <i>mediaBlocks</i>	16
2.7.3 <i>ToolStone</i>	17
2.7.4 <i>Illuminating Clay</i>	18
2.7.5 <i>Kontextbestimmung mittels Accelerometer</i>	19
2.7.6 <i>BlueWand</i>	20
2.7.7 <i>Würfel als TUI</i>	21
2.8 TangibleControl	23
3 Gestenerkennung	24
3.1 Begriffsbestimmung	24
3.2 Anforderungen an Gestenerkennungssysteme	25
3.3 Vorgehen bei der Gestenerkennung	26
3.4 Vergleich der Verfahren zur Gestenerkennung	28
3.5 Erkennung mittels Datenhandschuh	28
3.5.1 <i>Entwicklung der Datenhandschuhe</i>	28
3.5.2 <i>Anwendungen mit Datenhandschuhen</i>	30

3.6	Erkennung mittels Methoden der Computer Vision	31
3.6.1	Einteilung basierend auf der Vorgehensweise	32
3.6.2	Zeitliche Entwicklung und Anwendungen	33
4	Funktechnologien	35
4.1	WLAN (802.11)	35
4.1.1	Topologie	36
4.1.2	Funkschnittstelle	37
4.2	Bluetooth	37
4.2.1	Topologie	38
4.2.2	Technische Grundlagen	39
4.3	ZigBee	40
4.4	Proprietärer Funk	42
4.5	Infrarot	43
4.5.1	IrDA Data	43
4.5.2	IrDA Control	44
4.5.3	Protokollarchitektur	44
5	Technische Realisierung	45
5.1	Auswahl der Hardwarekomponenten	45
5.2	Sensorik	46
5.2.1	Auswahl – InertiaCube ²	46
5.3	Kommunikation	49
5.3.1	Auswahl – connectBlue OEM Serial Port Adapter 33i	50
5.4	Weitere Hardware	51
5.4.1	Akkumulatoren	51
5.4.2	Spannungsregler	52
5.5	Systemüberblick über die Hardware	52
5.5.1	Blockschaltbild	53
5.6	Anforderungen an die Software	54
5.7	Software für unterschiedliche Sensorik	55
5.7.1	Einbinden des InertiaCube ²	56
5.8	Einbindung der Gestenerkennung in ein Programm	56
5.8.1	„Hello world“ für Gestenerkennung	56
5.8.2	Konfigurationsmöglichkeiten	57
5.8.3	Konzept der Gesten	58
5.8.4	Unterteilung der Gesten	58
5.8.5	Hinzufügen und entfernen von Gesten	59
5.8.6	Log- und Visualisierungsmöglichkeiten	60
5.9	Überblick über das gesamte System	61
5.9.1	Verwendete Hardware	61
5.9.2	Notwendige Software	61
5.10	Eigenschaften verschiedener geometrischer Formen	62
5.10.1	Kegel	62
5.10.2	Pyramide	63
5.10.3	Quader	64
5.10.4	Würfel	64
5.10.5	Zylinder (groß)	65
5.10.6	Zylinder (klein)	65
5.10.7	Mögliche Interaktion mit den Artefakten	66
5.10.8	Möglichkeiten der Steuerung im Vergleich zu GUI	67
6	Szenarien / Beispielapplikationen	69

6.1 Visualisierung	69
6.1.1 VRML	69
6.1.2 Java3D	72
6.2 RobotControl	74
6.3 MediaplayerControl	76
6.3.1 Steuerung mittels Pyramide	78
6.3.2 Steuerung durch mehrere TUIs	81
7 Fazit und Ausblick	85
7.1 Weiterführende Arbeit	85
7.1.1 Hardware	85
7.1.2 Software	87
7.1.3 Empirische Untersuchungen	87
7.2 Fazit	87
Literatur	89
Lebenslauf	92
Eidesstattliche Erklärung	93

Abbildungsverzeichnis

Abbildung 1: Abbildung der Gesten auf Aktionen	1
Abbildung 2: Von GUI zu Tangible User Interface [22]	9
Abbildung 3: Konzept der Marble Answering Machine [18]	10
Abbildung 4: Strukturelle Darstellung der Möglichkeiten von Tangible Bits [22]	11
Abbildung 5: Dangling String: geringer (links) bzw. hoher(rechts) Netzwerkverkehr [55]	12
Abbildung 6: Schematischer Aufbau eines DigitalDesk [56]	15
Abbildung 7: Media sequencer für mediaBlocks [50]	16
Abbildung 8: ToolStone [43]	17
Abbildung 9: Illuminating Clay im Einsatz [40]	19
Abbildung 10: BlueWand IIx [1]	20
Abbildung 11: IMU Hardware [2]	21
Abbildung 12: Hardware in transparenten Kunststoffwürfel [52]	22
Abbildung 13: Prototyp des iCube [38]	22
Abbildung 14: Vorgehensweise bei der Gestenerkennung	26
Abbildung 15: Sayre Glove, DataGlove (v.l.n.r.)	30
Abbildung 16: Drahtloser Datenhandschuh mit Accelerometern	31
Abbildung 17: Kappe mit montierter Kamera bzw. Bild von dieser Kamera	33
Abbildung 18: Independent Basic Service Set [5]	36
Abbildung 19: Infrastruktur Modus: Basic Service Set (links), Extended Basic Service Set (rechts) [5]	36
Abbildung 20: Bluetooth Piconet	38
Abbildung 21: Bluetooth Scatternet	39
Abbildung 22: IEEE 802.15.4 Stack [8]	41
Abbildung 23: ZigBee Netzwerk Topologie [24]	41
Abbildung 24: Übersicht der ISM Frequenzbereiche	43
Abbildung 25: InterSense InertiaCube ²	48
Abbildung 26: InertiaCube ² mit gekürzten Kabel und neuem Stecker	49
Abbildung 27: connectBlue Serial Port Adapter 33i	50
Abbildung 28: Akkupack für die Hardware	51
Abbildung 29: Stecker der Akkupacks	52
Abbildung 30: Größenvergleich der Hardware	53
Abbildung 31: Schematischer Aufbau des Hardware Prototypen	54
Abbildung 32: Kegel mit Elektronik	63
Abbildung 33: Pyramide mit Elektronik	63
Abbildung 34: Würfel mit Elektronik	64
Abbildung 35: Großer Zylinder mit Elektronik	65
Abbildung 36: Kleiner Zylinder mit Elektronik	66
Abbildung 37: Visualisierung einer Bewegung mittels VRML Linienzug. Gesamte Bewegung (links) und Detail dieser Bewegung (rechts)	70
Abbildung 38: Detailansichten der Darstellung einer Bewegung mittels VRML	72
Abbildung 39: Visualisierung einer Bewegung mittels Java3D mit verschiedenen Perspektiven	73
Abbildung 40: Roboterarm in zwei unterschiedlichen Zuständen	75
Abbildung 41: Situation in einem Kaffeehaus	76
Abbildung 42: Liegende Pyramide mit Bewegungen	78
Abbildung 43: Stehende Pyramide mit Bewegungen	80
Abbildung 44: Schachtel und Dose mit Bewegungen	82

Tabellenverzeichnis

Tabelle 1: Gegenüberstellung der Vor- und Nachteile von Datenhandschuh, Methoden der Computer Vision und Sensorik in Artefakten [27][37]	29
Tabelle 2: Überblick über aktuelle lokale Funktechnologien	35
Tabelle 3: Wichtige WLAN Standards [29]	36
Tabelle 4: Leistungsklassen von Bluetoothgeräten	40
Tabelle 5: Überblick über verfügbare Sensoren. Einträge mit "-" wurden vom Hersteller nicht angegeben.	47

Das TUI soll es ermöglichen die Bewegungen und Gesten des Benutzers bzw. des eines Artefaktes zu erkennen. Dieses Interface soll in unterschiedliche alltägliche Gegenstände eingebettet werden können. Die Gesten können erkannt werden, wenn dieser „smarte“ Gegenstand bewegt wird. Die notwendigen Informationen wie etwa die erkannten Gesten soll ein API zur Verfügung stellen um, unabhängig von der darunter liegenden Technologie, Aktionen ausführen zu können.

Aus diesen Anforderungen ergeben sich zwei große Problemgruppen, die Hardware die für die Aufzeichnung der Bewegung mittels Sensoren verantwortlich ist und die Software mit deren Hilfe eine Abstraktion von den Sensordaten auf erkannte Bewegungen erfolgt. Die Software besteht aus einem API, welches es dem Anwendungsentwickler erlauben soll die Möglichkeiten der Gestenerkennung einfach in Anwendungen zu integrieren. Somit ergeben sich folgende Anforderungen an das API:

- Steuerungsmöglichkeit von Anwendungen mittels Handbewegungen.
- Einfache Integration der Bewegungserkennung in Anwendungen.
- Bereitstellen „natürlicher“ und intuitiver Bewegungen.
- Unabhängigkeit von der Form des Artefaktes.
- Erweiterungsmöglichkeit der Software für weitere Sensorklassen (z.B. Position).
- Bereitstellen der erkannten Gesten und aller relevanten Sensordaten.
- Freie Gestaltung von zusätzlichen Gesten und Bewegungen.
- Hinzufügen und entfernen von Bewegungen zur Laufzeit.
- Weitestgehende Abstrahierung des gesamten Hardwareteiles.
- Unabhängigkeit von der darunter liegenden Sensorik und der verwendeten Sensortechnologie.

Neben diesen Anforderungen an das API gibt es noch eine Reihe von Anforderungen und Einschränkungen an das Gesamtsystem, welche die Hardware aber auch mögliche Applikationen betreffen:

- Handbewegungserkennung ohne zusätzliche Infrastruktur wie Kameras, Netzwerke, usw..
- Autonomer Betrieb soll möglich sein.
- Geringe zeitliche Verzögerung zwischen dem Ausführen und dem Erkennen einer Bewegung.
- Gesamte Hardware (Sensor, Sender,...) muss bequem in/an der Hand Platzfinden.
- Geringes Gewicht und Größe der Hardware.
- Im autonomen Betrieb soll eine Laufzeit von zumindest einer Stunde erreicht werden.
- Konzipierung und Fertigung eines Prototyps der Hardware.
- Die Hardware soll flexibel in verschiedene geometrische Formen (Zylinder, Würfel, ...) mit unterschiedlicher Größe integriert werden können.
- Es sollen geometrisch unterschiedliche Artefakte erstellt werden.
- Untersuchung der Auswirkung der Form auf die Eingabemöglichkeiten.
- Keine zusätzliche Beeinträchtigung des Benutzers durch Handschuhe, Sensoren am Körper oder ähnliches.
- Evaluieren der Eingabemöglichkeiten (unterschiedliche bzw. unterscheidbare Befehle, Befehle pro Zeiteinheit).
- Abbildung der Bewegungen auf den Funktionsumfang der Beispielanwendungen.
- Entwurf von Implementierung von Beispielapplikationen.

Die gesuchten Beispielapplikationen sollen die Möglichkeiten des gesamten Systems leicht und verständlich darstellen. Diese Anwendung(en) soll(en) möglichst im Bereich von „Smart Objects“ liegen und als zusätzliche Motivation für die Realisierbarkeit von „Ubiquitous Computing“ allgemein und Tangible User Interfaces im Speziellen dienen.

Das System umfasst somit sowohl die Entwicklung und Erstellung der für die Anwendung notwendigen Hardware, als auch die Software mit den dazugehörigen Beispielanwendungen.

Das API soll es dem Anwendungsentwickler ermöglichen ohne Wissen über die Hardware und die verwendete Sensortechnologie die gesamte Bewegungserkennung zu nutzen. Hierbei soll jedoch versucht werden die Einschränkungen möglichst gering zu halten, sodass der Einsatz der Hardware und des APIs nicht auf bestimmte Formen, Bewegungen, Sensoren usw. beschränkt ist.

Die Anforderungen an die Hardware sind bereits sehr speziell. Da die Erkennung der Gesten ohne größere zeitliche Verzögerungen erfolgen soll, ist es nicht möglich die Sensordaten zwischenspeichern und nachträglich zu analysieren. Durch den zusätzlich gewünschten autonomen Betrieb ist eine kabelgebundene Übertragung der Daten und Stromversorgung der Sensorik nicht möglich. Als Lösung bietet sich eine batteriebetriebene Elektronik an, welche die aktuellen Sensordaten unmittelbar via Funk an eine Auswerteeinheit übermittelt.

Bereits vom autonomen Gerät erkannte Bewegungen müssen sowieso an ein übergeordnetes System weitergegeben werden, deshalb würde eine Recheneinheit die direkt an die Sensorik angeschlossen ist nur einen geringen Nutzen bringen. Diese Vorgehensweise würde jedoch einige Nachteile mit sich ziehen, wie etwa die Baugröße, das Gewicht und den erhöhten Stromverbrauch, wobei dieser wieder reduziert werden könnte, indem nur Daten gesendet werden sobald eine Bewegung erkannt wurde. Da man zusätzlich die Möglichkeiten bei der Bewegungserkennung durch die Rechenleistung dieser mobilen Recheneinheit (z.B.: Mikrocontroller) im Vorhinein einschränken würde, wurde von diesem Ansatz Abstand genommen.

Aufgrund der Anforderung, dass das System ohne zusätzliche Infrastruktur betrieben werden soll könnten stationäre Computer-Vision-Systeme nicht verwendet werden. Auch der Einsatz von zusätzlichen Sensoren im Raum um etwa die Position bestimmen zu können ist nicht möglich. Die Wahl der Sensorart ist dadurch schon beinahe vorgegeben, da Accelerometer und Gyroskope (Beschleunigungssensoren) ohne Infrastruktur auskommen und trotzdem Informationen über die Bewegung liefern. Diese Informationen beziehen sich jedoch meist nur auf die Orientierung im Raum und nicht auf die Position. Eine exakte Positionsbestimmung mittels Beschleunigungssensoren ist nur mit verhältnismäßig großem Aufwand zu erreichen. Die beiden zur Bewegungserkennung am häufigsten eingesetzten Sensoren, Kameras der Computer Vision und Datenhandschuhe sind aufgrund der Anforderungen nicht möglich bzw. nicht erwünscht.

Bezüglich Netzwerk- und Funktechnik kommen somit auch viele der nationalen und globalen Technologien nicht in die nähere Auswahl.

Ein weiterer Teil der Problemstellung betrifft die unterschiedlichen geometrischen Formen und die damit verbundene unterschiedliche Handhabung einerseits in der Software andererseits natürlich auch durch den Anwender. Dies kann etwa auch bei Kindern, die mit einer Kugel anders umgehen als etwa mit einem Würfel, beobachtet werden. Hierbei gilt es somit die Vorteile der jeweiligen Form zu nutzen. Dies ist natürlich speziell bei der Entwicklung der Beispielanwendungen von Interesse.

Das Hauptaugenmerk liegt nicht in der Implementierung von möglichst effizienten und schnellen Algorithmen zur Gestenerkennung. Verbesserte Verfahren mit detaillierterer

mathematischer Betrachtung usw. können somit eine signifikante Verbesserung der Gestenerkennung liefern.

1.3 Überblick

Eine grundlegende Einführung in die Thematik des Ubiquitous Computing ist in Kapitel 2 zu finden. Nach einer kurzen geschichtlichen Betrachtung werden neben der Vision und den Zielen auch Gründe gezeigt welche die Entwicklung von ubiquitären Anwendungen erlauben. Durch die genauere Betrachtung der wesentlichen Merkmale von Tangible User Interfaces werden der Fokus und das Ziel dieser Arbeit näher erläutert. Mithilfe des Abschnittes zum Thema Kontext kann die Bedeutung der Sensorik im Bereich des Ubiquitous Computing abgerundet werden. Abschließend werden besondere Beispiele aus dem Themengebiet vorgestellt, wobei die Auswahl mit einer Zentrierung auf die für diese Arbeit wesentlichen und ähnlichen Anwendungen getroffen wurde.

Kapitel 3 zeigt die beiden bisher gebräuchlichsten Ansätze zur Bewegungserkennung. Einerseits die Erkennung mittels Computer Vision, also die Auswertung von Kamerabildern, andererseits durch Datenhandschuhe. Hierbei werden die Möglichkeiten sowie die Vor- und Nachteile der beiden Ansätze erörtert. Neben einer geschichtlichen Betrachtung erfolgt auch eine kurze Darstellung bekannter Anwendungen. Weiters werden die Begriffe „Geste“ und „Gestenerkennung“ detaillierter betrachtet und Anforderungen an Gestenerkennungssysteme definiert.

Im Abschnitt 4 werden zentrale Eigenschaften wie etwa Frequenzbereich, Datenrate, Reichweite, Komplexität und Stromverbrauch von lokalen Kommunikationssystemen zur drahtlosen Datenübermittlung gegenübergestellt. Der Vergleich umfasst die Standards Bluetooth, IrDa, WLAN und ZigBee, zusätzlich wird noch auf proprietäre Technologien eingegangen.

Durch einen Überblick über die vom jeweiligen Verfahren unterstützten Topologie können auch die möglichen Einsatzgebiete näher definiert werden. Zusätzlich werden die Besonderheiten und die wesentlichen technischen Grundlagen der einzelnen Technologien aufgezeigt.

Abschnitt 5 befasst sich mit der technischen Realisierung. Zuerst erfolgt die Begründung für die Auswahl der Sensorik und eine entsprechende Beschreibung dieser, anschließend wird noch auf die gewählte Datenübertragung und die weiteren erforderlichen Hardwareteile eingegangen.

Der zweite große Teil beschäftigt sich mit der Software, dies umfasst einerseits die Funktionalität des entwickelten APIs und die daraus resultierenden Möglichkeiten und andererseits auch die Software die von Drittanbietern verwendet wird. Ein weiterer Punkt in diesem Kapitel beschäftigt sich mit den sechs verschiedenen geometrischen Formen die auf ihre Eignung als TUI untersucht wurden.

Die realisierten Anwendungen und dazugehörigen Szenarien werden in Kapitel 6 behandelt. Zu Beginn werden zwei Visualisierungsmöglichkeiten der Sensorwerte vorgestellt die direkt auf der Funktionalität des APIs beruhen, eine weitere Visualisierung zeigt eine Anwendung in einem industriellen Umfeld.

Als Hauptanwendung wurde ein Mediaplayer realisiert. Hierbei ist eine Steuerung der wesentlichen Funktionen mit zwei verschiedenen TUIs möglich, da die Software und der Funktionsumfang an die Eigenschaften der verschiedenen Formen angepasst wurden. Hierbei werden für die Steuerung bis zu 17 unterschiedliche Bewegungen verwendet.

Abschließend erfolgt in Abschnitt 7 eine Bewertung und eine Aussicht. Hierbei werden Ansätze zur Weiterentwicklungen der Hardware und der Software sowie Möglichkeiten für empirische Studien erläutert. Bei der Hardware betreffen die Änderungen unter anderen zusätzliche Ein- und Ausgabemöglichkeiten. Im abschließenden Fazit werden die Resultate kritisch bewertet.

2 Ubiquitous Computing

Unter Ubiquitous Computing werden allgegenwärtige in die Umgebung integrierte Recheneinheiten verstanden. Speziell in Europa wird hierfür mittlerweile verstärkt der Begriff Pervasive Computing verwendet.

2.1 Entstehung von Ubiquitous Computing

Bereits 1927 schrieb Heidegger wie der Mensch in und durch die Umwelt lernt. Durch das Angreifen der Gegenstände gelingt es uns ein tieferes Verständnis zu erlangen. Weiters ist an Heidegger bemerkenswert, dass er der Meinung war, dass die moderne Technik alles durchdringen wird. Dies könnte nun, fast einhundert Jahre später, wirklich eintreten, wobei Heidegger die tatsächlichen Auswirkungen und Technologien für die Realisierung natürlich nicht voraussagen konnte. Er ahnte jedoch, dass es eine einmalige Chance für die Technik wird, sobald sie allgegenwärtig ist, hierdurch gelingt es die Menschen genau dort zu erreichen wo sie sind: in ihrer gewohnten Umwelt.

1991 führte Weiser in „The Computer for the 21st Century“ Heideggers Idee von der all-durchdringenden Technik fort, wengleich die Betrachtung der Technik gegenüber wesentlich freundlicher ausfiel als dies noch bei Heidegger der Fall war. Weiser beginnt seinen Artikel mit folgenden Zeilen:

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.” [54]

Hier spricht er von den verschwindenden Technologien, die sich in die Struktur des täglichen Lebens integrieren sollen, bis sie nicht mehr als eigene Technologie erkannt werden können. Dieser Ansatz wurde zum zentralen Punkt eines neuen Zweiges in der Informatik. Weiser prägte hierfür als Erster den Begriff des *Ubiquitous Computing*, also der allgegenwärtigen Rechner.

Die Grundidee dahinter erklärte er anhand eines einfachen Beispiels, der Schrift. Das Schreiben kann in den Industrieländern als ubiquitäre Technologie angesehen werden. Schrift findet sich nicht nur in Büchern und Zeitschriften, sondern auch auf Plakaten, Straßenschildern usw.. Die Schrift ist somit im Hintergrund verfügbar, es ist jedoch die Aufmerksamkeit nicht erforderlich. Das Geschriebene wird einfach auf einen Blick erkannt, ohne das man sich jedoch bewusst ist gerade zu lesen.

Die Vision die dahinter steckt, ist ein Paradigmenwechsel im Umgang mit dem (Personal) Computer. War es vor Jahrzehnten noch üblich, dass ein Computer von vielen Leuten genutzt wurde, so hat sich dies hin zum Personal Computer geändert, also ein Computer für jeden. Diese Entwicklung wird jedoch verstärkt weitergehen, sodass eine Person zukünftig mit vielen Computern interagieren wird, wobei der Begriff „Computer“ hier allgemeiner zu sehen ist. Es kann jeder Mikrocontroller oder sonstiges Gerät mit Rechenleistung sein. Dieser Paradigmenwechsel in der Nutzung erfordert jedoch auch eine geänderte Form der Interaktion mit den Computern.

Ziel des Ubiquitous Computing sind allgegenwärtige, in der Umgebung integrierte Computer in der Form von vielen kleinen, vernetzten, autonomen Mikroprozessoren. Die erforderliche Hardware wird in die räumliche Umgebung hineinverlegt, Rechenleistung in Gegenstände und Dinge wie Fußböden, Wände, Möbel und Notizblöcke integriert, das Interface als

Funktionserweiterung in Artefakten des täglichen Lebens integriert. Die Umgebung registriert, erkennt und beobachtet ihre Bewohner, reagiert auf sie, antizipiert gar ihre Wünsche und Handlungen. Die Computer werden zu einem integrierten, unsichtbaren Teil des Alltags und verschwinden aus der bewussten Aufmerksamkeit. Anstelle eines privaten, universellen Gerätes gibt es eine Vielzahl unabhängiger, spezialisierter Objekte, die von jedem benutzt werden können. [18]

2.2 Gründe, die Ubiquitous Computing ermöglichen

Obwohl Weiser bereits 1991 von Ubiquitous Computing sprach entfaltete sich dieser Zweig in den ersten Jahren nur sehr zaghafte. Das lag jedoch primär daran, dass die technischen Grundlagen noch nicht erfüllt waren. Mattern [30] spricht von fünf Gründen, welche die weitere Entwicklung von ubiquitären Anwendungen erlauben:

- *Moore's Law*: Dieser Punkt umfasst den gesamten Bereich der Halbleiterentwicklung. Recheneinheiten und Speicher werden immer schneller, kleiner und preiswerter. Die Möglichkeiten der mobilen Stromversorgung steigen jedoch nicht in diesem Ausmaß. Aufgrund dieser Verbesserungen werden eine weite Verbreitung und die Integration in die Umwelt bzw. in alltägliche Objekte ermöglicht.
- *Neue Materialien*: Es gibt ständig Weiterentwicklungen in diesem Bereich durch die etwa Displays kleiner und kontrastreicher werden. Leiterplatten werden biegsam und können z.B. auch in Kleidung integriert werden oder die erforderlichen Leitungen werden mit dem Stoff verwoben. Als weiteres Beispiel kann die Forschung auf dem Bereich der elektronischen Tinte und dem dazugehörigen Papier genannt werden.
- *Fortschritt der Kommunikationstechnologien*: Erst durch eine preisgünstige und bezüglich der Bauform kleine Datenübertragung wird es möglich die Daten an die gewünschten Stellen zu bringen. Dazu gehören jedoch nicht nur Funktechnologien sondern auch Technologien, die andere Träger zur Übermittlung, wie etwa den Menschen oder Baumaterialien im Gebäude verwenden.
- *Neue (Software-) Konzepte*: Als Beispiel kann hier etwa die spontane Vernetzung genannt werden, also das Geräte automatisch voneinander wissen wenn ein Anderes neue Informationen hat oder geographisch in die Nähe kommt.
- *Bessere Sensoren*: Durch kleinere, schnellere, exaktere und preiswertere Sensoren können leichter Informationen über die Umwelt aufgenommen werden. Dadurch wird eine genauere und umfangreichere Kontextbestimmung ermöglicht. Die Bedeutung der Informationen über den Kontext werden in Kapitel 2.5 erläutert.

2.3 Einordnung der Virtualitäts- und Realitätsbegriffe

Weiser grenzt Ubiquitous Computing sehr stark von „Virtual Reality“ ab. Er bezeichnet die beiden Ansätze sogar als gegensätzlich. In Virtual Reality wird versucht durch den Computer eine eigene Welt zu generieren in die dann der Benutzer integriert wird. Da die Gegenstände nicht real sind ist auch die Interaktion des Benutzers mit der virtuellen Welt schwierig und gewöhnungsbedürftig. Hinzu kommt noch, dass meist eine spezielle Ausrüstung, wie Datenhandschuhe oder spezielle Brillen, für den Benutzer notwendig ist um in der virtuellen Welt teilnehmen zu können.

Bei Virtual Reality wird versucht die Welt mit großem Aufwand zu simulieren anstelle die Realität mit Informationen anzureichern, somit werden jedoch Tische, Büros, das Wetter, Bäume oder auch Personen ohne der notwendigen Ausrüstung aus dieser Welt ausgeschlossen.

Ein Vorteil von Virtual Reality liegt darin, dass auch Bereiche untersucht werden können, die dem Menschen sonst nicht oder nur unter großer Gefahr zugänglich sind. Dieses Gebiet kann mit Ubiquitous Computing nicht abgedeckt werden. [54]

2.3.1 Embodied Virtuality

Weiser führt auch den Begriff der „Embodied Virtuality“ ein:

“ubiquitous, invisible computing ist so strong that some of us use the term “embodied virtuality” to refer to the process of drawing computers out of their electronic shells. The “virtuality” of computer-readable data [...] is brought into the physical world” [54]

Die Idee dahinter ist somit die reale Welt mit digitalen Daten anzureichern. Es kann also von einer leicht abgeschwächten Form von Ubiquitous Computing gesprochen werden, wobei eine strikte Trennung der Gebiete kaum möglich ist.

2.3.2 Augmented Reality

Augmented Reality versucht die Frage zu beantworten, wie man digitale Medien in die reale Welt integrieren kann.

Der am häufigsten verwendete Ansatz ist das Einblenden beziehungsweise die Überlagerung von digitalen Informationen in eine reale Umgebung oder ein Bild von ihr, mittels head mounted displays (HMD) – eine Art Brille auf der die Daten eingeblendet werden – oder Video Projektoren. [22]

Der Begriff „Augmenting“ wird von McGee, Cohen und Wu wie folgt definiert:

“Adding something to a real world object to cause it to represent, denote or be associated with something else.” [32]

Hierbei erfolgt keine Einschränkung auf rein visuelle Erweiterungen. Eine wenn nicht sogar die bahnbrechende Arbeit auf dem Gebiet der Augmented Reality ist der DigitalDesk von Wellner (siehe Kapitel 2.7.1).

2.3.3 Mixed Reality

Als Oberbegriff wird häufig der von Milgram bereits 1994 geprägte Begriff „Mixed Reality“ [33] verwendet, womit allgemein jede Art der Mixtur realer und virtueller Objekte bezeichnet wird. Unter Mixed Reality fallen unter anderem Embodied Virtuality, Augmented Reality, Tangible User Interfaces sowie Augmented Virtuality, also Virtual Reality Umgebungen mit einzelnen greifbaren Elementen, welche die Interaktion intuitiver und realistischer machen. [18]

2.4 Tangible User Interface (TUI)

2.4.1 Graspable User Interfaces

Als Vorgänger des Tangible User Interface kann das Konzept der „Graspable Interfaces“ von Fitzmaurice genannt werden. Er definiert in [12] Graspable User Interfaces als Systeme, die graphische Repräsentationen mit Hilfe angreifbarer Griffe (handle) manipulierbar machen. Fitzmaurice beschreibt fünf grundlegende Eigenschaften von Graspable User Interfaces und untersuchte insgesamt rund 15 Systeme inwieweit sie diese fünf Kriterien erfüllen.

Das zentrale und essentielle Merkmal ist das so genannte *spatial multiplexing*, das man als räumliche Vervielfachung oder auch als räumliche Verteilung übersetzen kann. Die wesentlichen Punkte der fünf Eigenschaften sind:

- Räumliches Verteilen, d. h. dass jede Funktion festen, funktionsbezogenen Ein- oder Ausgabegeräten zugeordnet ist, die jeweils einen eigenen Platz im Raum einnehmen. Ein- und Ausgabe verteilen sich gleichermaßen im Raum. Dies ermöglicht die voneinander unabhängige, gleichzeitige und anhaltende Auswahl von Objekten.
- Die Gleichzeitigkeit (der Benutzung, der Sichtbarkeit der Objekte)
- Die Verwendung „starker“, spezifischer Werkzeuge von eindeutiger Form
- Die Verwendung räumlich orientierter bzw. wahrnehmbarer Geräte (Position, Orientierung oder Nähe zu anderen Geräten)
- Die räumliche Konfigurierbarkeit und Anordnungsmöglichkeit der Arbeitsgegenstände im Arbeitsbereich

Die räumliche Verteilung ist laut Fitzmaurices experimenteller Untersuchung besonders effektiv, da sie eingeübte alltägliche motorische Bewegungen und die Hand-Auge Koordination ausnutzt. [12][18]

2.4.2 Von GUI zu TUI

Der Begriff Tangible User Interface mit der damit verbundenen Bedeutung wurde von Ishii und Ullmer 1997 geprägt. In diesen Zusammenhang wurden auch *Tangible Bits* eingeführt. Das Ziel dieser ist es den Spalt zwischen dem virtuellen Raum, der Umwelt und den vorder- und hintergrundigen Aktivitäten der Menschen zu überwinden.

Viele sehr spezialisierte Geräte, wie z.B. zur Zeitmessung, zum Zeichnen geometrischer Figuren, zur Unterhaltung usw. wurden durch ein Universelles ersetzt, den Personal Computer. Durch diese PC-zentrierte Arbeits- und Sichtweise wird die gesamte Information mittels einer graphischen Benutzeroberfläche auf einer rechteckigen Fläche dargestellt. Als Eingabegerät stehen primär Maus und Tastatur zur Verfügung. Mit diesen beiden sehr generellen Geräten sollen nun all diese spezialisierten Geräte bedient werden können, die durch den PC abgelöst wurden. Dies verursacht zwangsweise Probleme.

Die Grundidee hinter Tangible User Interface ist es die uns umgebende Welt mit digitalen Informationen anzureichern indem die Informationen an beliebige physikalische Artefakte gekoppelt werden. Es sollen somit greifbare Gegenstände zur *Repräsentation* und *Manipulation* der *digitalen Daten* entstehen. Die Idee der Änderung von GUI zu TUI mit der damit verbundenen Änderung der Interaktion ist in Abbildung 2 dargestellt.

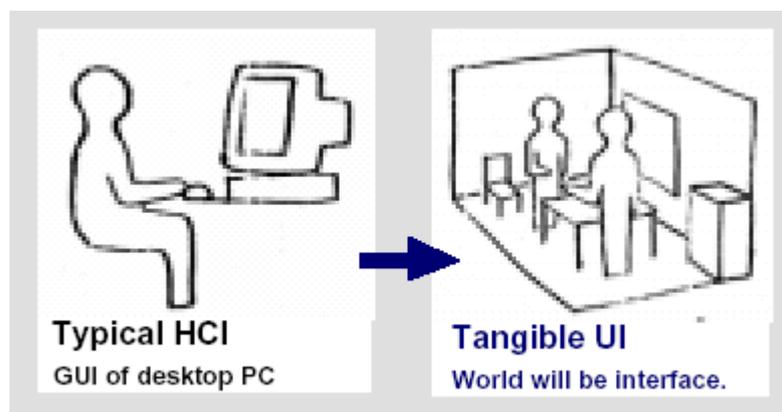


Abbildung 2: Von GUI zu Tangible User Interface [22]

Der Wandel von GUI zu TUI bringt einige viel versprechende Änderungen für den Benutzer mit sich. Die Schnittstelle ist nicht mehr auf eine zweidimensionale, rein graphische Form beschränkt, sondern ist dreidimensional und vor allem haptisch-taktil erfahrbar. Gegenstände die sich zuvor auch schon im Raum befunden haben dienen nun zusätzlich als Repräsentation und zur Steuerung von digitalen Informationen. Der TUI-Ansatz gewinnt zusätzlich an Bedeutung da die Menschen nur aus der Tasterfahrung heraus lernen wie visuelle Eindrücke zu interpretieren sind, somit wurde bereits gelernt wie man mit den physikalischen Objekten umgehen kann und diese Erfahrung kann nun in der Manipulation von digitalen Informationen angewandt werden. Zusätzlich fällt die fehlende Verknüpfung zwischen der digitalen und der realen Welt weg, welche bisher oft ein Verständnisproblem verursacht hat, da man zwischen diesen Welten hin- und hergerissen war.

Die Sichtbarkeit der Artefakte und der damit verbundenen, im Vorhinein bereits bekannten, möglichen Aktionen sind wesentlich für Tangible Interfaces. Unsichtbar, wie es bei Ubiquitous Computing gefordert wird, sollen sie nur in dem Sinne sein, dass sie nicht aufdringlich sind und die dahinter liegenden Technologien nicht gesehen werden soll. Hier sieht man wieder die parallelen zur Schrift. Auch die ist normalerweise, bis auf spezielle Reklame, unaufdringlich im Hintergrund. Die Art wie die Schrift erzeugt oder angebracht wurde ist für den Benutzer auch hier nicht relevant. [18][22]

Die Marble Answering Machine von Bishop wird häufig als Paradebeispiel und Inspiration für TUIs genannt. Die Designskizze (Abbildung 3) wurde bereits 1992 angefertigt und ist eine der ersten, wenn nicht sogar die ersten, ausformulierten Visionen eines Tangible User Interfaces, später folgten auch Prototypen. Jeder Anruf der beim Anrufbeantworter einlangt wird durch eine farbige Murmel die in eine Schale rollt dargestellt. Legt man eine der Murmeln in eine kleine Mulde des Anrufbeantworters, so wird der dazugehörige Anruf abgespielt. Zusätzlich kann eine Murmel auch noch auf das Telefon gelegt werden, damit wird die anrufende Person zurückgerufen.

Die Marble Answering Machine erfüllt somit die beiden wesentlichen Eigenschaften von TUIs, die Repräsentation indem die digitale Information (der Anruf) durch ein physikalisches Objekt (eine Murmel) dargestellt wird und die Manipulation, wenn eine Murmel an eine bestimmte Position bewegt wird. [9][18]



Abbildung 3: Konzept der Marble Answering Machine [18]

2.4.3 Tangible Bits

Ishii und Ullmer versuchen mit den Tangible Bits die virtuelle und die reale Welt näher aneinander zu bringen indem die beiden Welten verknüpft werden. Hierzu werden drei mögliche Wege aufgezeigt [22] um Bits durch die Umwelt greifbar zu machen. Diese drei zentralen Konzepte sind:

- *Interaktive Oberflächen*: Jede Oberfläche innerhalb des Raumes (z.B. Wände, Decke, Türen, Fenster) wird zu einer aktiven Fläche zwischen der physischen und der virtuellen Welt.
- *Kopplung von Bits und Atomen*: Nahtlose Verknüpfung von alltäglichen physikalischen Objekten (z.B. Bücher, Karten, Gläser) mit der dazugehörigen digitalen Information.
- *Umgebende Medien*: Verwendung von Medien wie Musik, Licht, Luftzug als im Hintergrund liegende Schnittstelle zum virtuellen Raum die durch den Menschen nur peripher wahrgenommen wird.

Eine strukturelle Darstellung und Anordnung dieser drei Konzepte ist in Abbildung 4 ersichtlich.

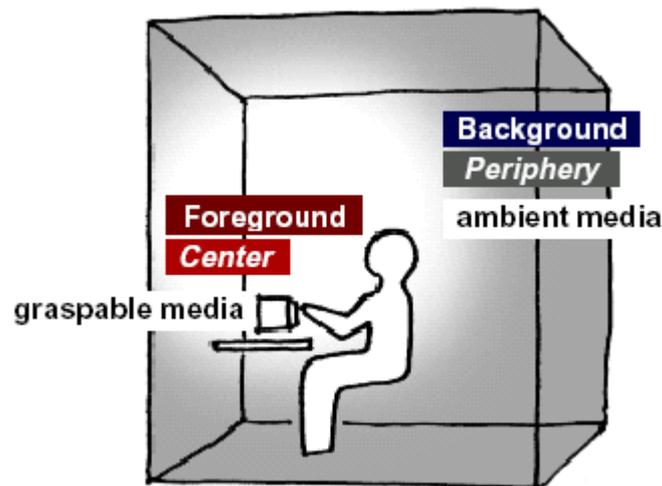


Abbildung 4: Strukturelle Darstellung der Möglichkeiten von Tangible Bits [22]

Der Großteil der Problemstellung aus Kapitel 1.2 lässt sich somit eindeutig dem zweiten Punkt, der Kopplung von Bits und Atomen, zuordnen, auch wenn noch weitere Anforderungen an das Gesamtsystem gestellt werden, welche diese Klassifizierung nicht direkt betreffen.

Zu den ersten beiden Konzepten werden in Kapitel 2.7 einige typische Beispiele näher erläutert.

Der Dritte Punkt, die Beeinflussung der umgebenden Medien, scheint am schwierigsten realisierbar zu sein, zumindest ist die Anzahl der Anwendungen, die zu dieser Gruppe zu zählen sind, am niedrigsten. Als Inspiration für diese Gruppe hat das „Dangling String Projekt“ von Jeremijenko gedient. Hierbei handelt es sich um eine Kunststoffschnur die an einem kleinen Motor hängt der an der Decke befestigt ist. Der Motor ist mit einem Ethernetkabel verbunden, sodass jedes Bit der Datenübertragung im Netzwerk eine kleine Bewegung des Motors hervorruft. Abhängig vom Netzwerkverkehr entstehen neben der visuellen Ausgabe auch akustische Signale, welche unterbewusst, und somit nicht störend, in mehreren Büros vernommen werden können. Auch wenn die Ausgabeform der Information sehr unscharf ist und interpretiert werden muss, so ist es trotzdem ein Gewinn an Information der bei Bedarf wahrgenommen wird. Der Dangling String ist in Abbildung 5 mit unterschiedlicher Netzwerkbelastung zu sehen. [55] Im linken Bild bewegt sich der Faden kaum, da eine niedrige Netzwerkbelastung vorherrscht, im rechten Bild dreht sich der Faden aufgrund des Netzwerkverkehrs entsprechend schneller.



Abbildung 5: Dangling String: geringer (links) bzw. hoher(rechts) Netzwerkverkehr [55]

2.4.4 Abgrenzung von TUI zu andern Konzepten

Tangible User Interfaces werden immer wieder als Teil anderer Konzepte verwendet, somit ist eine Trennung dieser Gebiete von Interesse. Speziell die signifikanten Eigenschaften bzw. die Unterschiede zu Augmented Reality und Ubiquitous Computing im Allgemeinen sollen hier hervorgehoben werden.

Während Ubiquitous Computing rechnerbasierte Fähigkeiten in die Umgebung menschlicher Arbeit hineinverlegen will, geht der Ansatz der Tangible Interfaces einen Schritt weiter und verkörpert die Rechenleistung. Die verkörperte Interaktion die dadurch möglich wird, fügt die Erkenntnis hinzu, dass die Repräsentationen ebenfalls Artefakte sind, welche manipuliert werden können. Der „Unsichtbarkeits“- Aspekt des Ubiquitous Computing kann bei den TUIs insofern anders ausgelegt werden, da die Artefakte nicht unsichtbar werden sollen bzw. können. Hierbei soll die Technologie die notwendig ist, um aus Objekten TUIs zu machen, unsichtbar sein. [18]

Viele Augmented Reality Systeme tendieren dazu, die visuellen Erweiterungen als reinen Zusatz beziehungsweise als ein Anhängsel zur realen Welt zu behandeln. Es wird eher Information zur Verfügung gestellt, durch die navigiert werden kann, als dass eine interaktive Kreation oder Veränderung ermöglicht würde. Die virtuellen Repräsentationen der Augmented Reality sind meist statisch. TUIs betonen im Vergleich stärker die Manipulation und weniger die Navigation, sie können jedoch durchaus ein Bestandteil von Augmented Reality Systemen sein. [18]

2.5 Kontext

Weiser ist bereits 1991 der Überzeugung, dass das Wissen über den Kontext eine der zentralen Fragen des Ubiquitous Computing darstellt.

„We have found two issues of crucial importance: location and scale. Little is more basic to human perception than physical juxtaposition, and so ubiquitous computers must know where they are.“ [54]

Auf die Größe der Geräte wurde bereits in Kapitel 2.2 kurz eingegangen, deshalb wird hier nur die Position bzw. allgemeiner der Kontext in dem sich ein Gerät befindet erläutert. Neben dieser sehr allgemeinen Begründung von Weiser, warum das Wissen über den Kontext wichtig gibt, existieren noch weitere, speziellere Gründe, warum Kontext Informationen für ubiquitäre Anwendungen von Bedeutung sind:

- Computer werden in mobilen Szenarien und Umgebungen abseits des klassischen Arbeitsplatzes verwendet.
- Rechenleistung wird in alltägliche Artefakte integriert.
- Es besteht nicht mehr die Absicht einen Computer zu bedienen. Benutzer führen Aktionen aus indem sie ein Objekt verwenden, ohne zu bedenken, dass sie einen Computer verwenden.
- Die Bedienung erfolgt von der breiten Masse und nicht nur noch von Experten.
- Effektivität ist nicht mehr das einzige Ziel, es zählt auch das Erlebnis des Benutzers. [44]

Unter Betrachtung der Visionen, Einschätzungen und Vorhersagen bezüglich der Datenverarbeitung der Geräte, der Infrastruktur und die Mensch-Maschine Kommunikation wird es ersichtlich, dass

- Kontext verfügbar ist und eine große Aussagekraft über ubiquitäre Umgebungen hat.
- Erwartungen und Erfahrungen von Benutzern im direkten Zusammenhang mit dem Kontext stehen.
- das Erfassen, das Darstellen, das Bereitstellen und die Verwendung von Kontextinformationen eine entscheidende Technologie ist, die es ermöglicht die Vision von unsichtbaren Computern in alltäglichen Geräten zu realisieren. [44]

Daraus resultiert nun die Frage nach dem Umfang des Kontextes und auch danach, wie der Kontext bestimmt bzw. erfasst werden kann.

Die Erfassung erfolgt mittels Sensoren, wobei der dafür notwendige Aufwand und die damit verbundenen Kosten von vielen Parametern abhängig sind. Der grundlegende Punkt ist die Art der Kontextinformation die erfasst werden soll. In [44] beispielsweise wird von über 130 Kategorien gesprochen, wenngleich viele der Kategorien sehr speziell sind und somit für Ubiquitous Computing keine Rolle spielen. Die Größen, die für ubiquitäre Anwendung von großer Bedeutung sein können sind, hauptsächlich Licht, Farben und Bilder, Töne, Bewegungen und Beschleunigung, Position, magnetische Feldstärke und Orientierung, Temperatur, Luftfeuchtigkeit, Luftdruck, Gewicht, Bewegungserkennung, Gas Sensoren und Bio-Sensoren.

Hier gibt es natürlich wieder eine Unmenge von verschiedenen Sensoren welche die einzelnen Gebiete abdecken. Die Wahl hängt somit immer davon ab, welche Anforderungen von Seiten der Kontextgewinnung an den Sensor gestellt werden. Die unterschiedlichen Eigenschaften der Sensoren sind beispielsweise Genauigkeit, Wertebereich, Preis, Größe, Gewicht, Robustheit und Stromverbrauch.

Alleine durch das Wissen über den Kontext kann noch nicht von einem intelligenten Gerät gesprochen werden. Der gesamte Bereich der die künstliche Intelligenz abdeckt ist mit Vorsicht zu genießen, aus diesem Grund verwendet Mattern [31][30] den Begriff „smart“, also gewieft, schlau oder auch klug. Smarte Objekte besitzen ein kontextsensitives Verhalten, zusätzlich können sie sich an sachbezogene Ereignisse erinnern, sie verfügen also über eine Speichermöglichkeit. Weiters besteht die Möglichkeit der Kommunikation zwischen smarten Objekten, das bedeutet, dass sie mittels eines Netzwerkes verbunden sind und somit bei Bedarf auf andere Objekte reagieren können.

Für Tangible User Interfaces ist das Wissen über den Kontext von existenzieller Bedeutung, sie vertrauen sozusagen auf die Verfügbarkeit von Informationen über die Situation in der sie verwendet werden. Das bedeutet jedoch natürlich nicht, dass die gesamten Kontextinformationen vorliegen müssen. Vielmehr geht es auf eine spezielle Eigenschaft der TUIs

zurück, und zwar der Darstellung von digitalen Daten. Um diese digitalen Informationen repräsentieren zu können, müssen Daten über den Zustand des Artefaktes oder über die Umgebung vorliegen. Es muss hier nicht erforderlich sein, das z.B. die genaue Position bekannt ist, sondern es kann der Raum in dem sich das TUI befindet genügen, somit ist klar welches Gerät damit gesteuert werden soll, oder es ist bekannt welche Person sich in der Nähe aufhält, es müssen somit nicht immer absolute Werte sein, sondern relative Werte können ebenso das erforderliche Wissen über den Kontext darstellen. Generell kann gesagt werden, dass die Positionsinformationen derzeit die größte Rolle spielen und am häufigsten verwendet werden.

Es kann festgehalten werden, dass die Kopplung zwischen digitaler und realer Welt nur möglich ist wenn Informationen über den Kontext vorliegen.

Die Kontextinformation des im Zusammenhang mit dieser Arbeit angefertigten Prototyps eines TUIs beschränkt sich auf eine Größe und zwar auf die Orientierung im Raum. Das bedeutet, dass die Neigung des Objektes bezüglich aller drei Freiheitsgrade bekannt ist. Somit können alle Rotationsbewegungen detektiert werden, reine Translationsbewegungen bewirken keine Änderung der Orientierung des Objektes und können nicht erkannt werden. Die Bewegungserkennung beschränkt sich auf die Erkennung von Drehungen, wobei diese üblicherweise einen Teil einer komplexeren Bewegung aus Rotation und Translation ausmacht.

Wenn auf einem Tisch ein Glas steht, wobei das Glas natürlich bereits mit dieser Bewegungserkennung ausgestattet wurde, und dieses am Tisch verschoben wird, so kann dies nicht erkannt werden. Sollte das Glas nun jedoch umgestoßen werden, so wird diese Bewegung aufgrund der Drehung um zumindest einen Freiheitsgrad erkannt. Da bei der Aufzeichnung der Werte auch noch weitere Informationen über die Bewegung gesammelt werden können, wie etwa der Zeitverlauf bzw. die Geschwindigkeit der Änderungen, kann auch zwischen dem Umstoßen des Glases und dem normalen Trinken unterschieden werden. Aufgrund der Neigung des Glases beim Trinken kann, abhängig vom Benutzer und der Form des Glases, auf den verbleibenden Inhalt geschlossen werden. Somit kann der Kellner informiert werden sobald das Glas leer ist. Der Kellner kann natürlich auch beim Umstoßen des Glases benachrichtigt werden.

Das zuvor beschriebene Beispiel beschäftigte sich primär mit dem Bewegungserkennungsaspekt und nicht mit den Möglichkeiten der Repräsentation und Manipulation von digitalen Informationen die bei TUIs auch gegeben sind. So könnte die Ausrichtung des Aufdrucks auf einem Glas (zum Benutzer oder vom Benutzer wegzeigend) eine gewisse digitale Information darstellen. Durch das Drehen des Glases könnte man dann diese Daten verändern.

2.6 Multimodale Interaktion

Unter *Multimodalität* versteht man die Verwendung mehrerer Sinneskanäle um Informationen zu übermitteln. Hierbei können alle Sinne, wie Sehen, Hören, Fühlen Tasten, Schmecken und der Gleichgewichtssinn einen dieser Sinneskanäle stellen. Zusätzlich gibt es noch die Unterscheidung, ob die Kanäle sequenziell oder parallel genutzt werden, abhängig davon sind auch die daraus resultierenden Möglichkeiten.

In einem ersten Ansatz wurde nur ein Kanal zur Kommunikation zwischen Mensch und Maschine verwendet, hierzu wurde etwa Tastatur, Maus oder ein Monitor eingesetzt. Durch die Verwendung mehrerer Kanäle wird versucht die Bandbreite zwischen Benutzer und Maschine zu erhöhen um die Interaktion effizienter zu gestalten. Als Erstes haben sich auditive Ausgaben neben den bereits existierenden visuellen Ausgaben etabliert. Eine weitere

Entwicklung betrifft die Eingabe mittels Sprache, wobei hier die Eingabe von wenigen Kommandos bis zur vollständigen natürlichen Sprache reichen kann.

Die Idee ist durch die Verwendung mehrerer Modalitäten die Kommunikation schneller, intuitiver, natürlicher und einfacher zu gestalten. Andererseits kann eine gezielte Redundanz auch zu einer Reduktion von Fehlern führen oder es kommt zu einer Reduktion der Belastung der einzelnen Sinne. Weiters kann durch die Verwendung mehrerer Modalitäten auch eine parallele Bearbeitung von Daten erreicht werden.

Eine weitere wesentliche Modalität die sich in diesen Zusammenhang immer mehr herauskristallisiert ist die Gestenerkennung. Diese Form punktet speziell im Bereich der Natürlichkeit, der intuitiven Eingabe und in der Parallelität. Ein Vorteil liegt darin das die Gesten oft sogar unterbewusst durchgeführt werden und kein zusätzlicher Aufwand für den Benutzer vorhanden ist. Die erkannten Gesten können zusätzlich zu den anderen Modalitäten als Eingabe verwendet werden und entsprechend die Möglichkeiten der Interaktion erweitern.

2.7 Beispielhafte Anwendungen

Im Laufe der Zeit wurden einige für Ubiquitous Computing sehr typische Anwendungen entwickelt, wobei man beobachten kann, dass der Fortschritt von den Weiterentwicklungen in den einzelnen Teildisziplinen wie Bewegungserkennung, Elektronik, Fertigungstechnik usw. getrieben wird.

Weiters muss man festhalten, dass sich die Anwendungen noch in einem sehr frühen Stadium befinden und kaum über den Status des Prototypen hinausgehen, folglich ist auch die Anzahl der industriell gefertigten Systeme noch sehr gering.

2.7.1 DigitalDesk

Weiser zeigte bereits 1991 die ersten Anwendungen des DigitalDesks, dessen schematischer Aufbau in Abbildung 6 zu sehen ist. Das System besteht im Wesentlichen aus drei Komponenten: einer Kamera, einem Projektor und einem System zur Datenverarbeitung. Mithilfe der Kamera kann die Bewegung der Hand verfolgt werden, zusätzlich können beispielsweise Skizzen die der Benutzer gezeichnet hat aufgenommen werden. Durch den Projektor können auf einer beliebigen Position in der Szene zusätzliche digitale Informationen eingeblendet werden. Der Computer ist für die Bewegungserkennung zuständig und generiert die Daten für den Projektor.

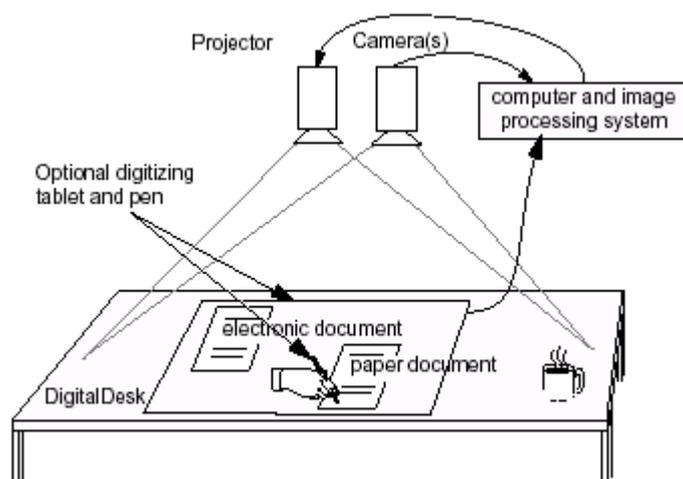


Abbildung 6: Schematischer Aufbau eines DigitalDesk [56]

Ein Ziel des Ansatzes, welcher zu den Augmented Reality Systemen zu zählen ist, ist es, die Vorteile der papierbasierten und der digitalen Welt zu vereinen, indem die reale Welt mit zusätzlichen digitalen Informationen angereichert wird. So können etwa die Geschwindigkeit beim Zeichnen von Skizzen und die allgemeine Akzeptanz und Vertrautheit als Vorteile von Ansätzen mit Papier gewertet werden. Wobei hingegen das Suchen innerhalb von Texten oder das Kopieren von Grafiken als Vorteil der digitalen Version zu sehen ist.

Die Bearbeitung der Dokumente kann sowohl in der digitalen als auch in der realen Welt stattfinden, d.h. es kann beispielsweise auf dem Papier eine Grafik gezeichnet werden, danach wird ein Bereich markiert der kopiert werden soll. Die Kopie wird dann durch den Projektor an der gewünschten Stelle eingeblendet.

Neben der Bearbeitung von gewöhnlichen Dokumenten gibt es noch weitere Anwendungsbeispiele, so kann etwa ein Taschenrechner, der nur auf dem Tisch projiziert wird, mit einem Finger bedient werden. Zusätzlich können auch Zahlen aus anderen virtuellen oder realen Dokumenten direkt in den Taschenrechner übernommen werden.

Weiters ist die Kopplung von zwei DigitalDesks möglich, hierbei werden die erforderlichen Daten vom jeweils anderen Schreibtisch durch den Projektor eingeblendet. Diese Daten können von beiden Personen bearbeitet werden. [56][57]

2.7.2 mediaBlocks

Die Grundidee hinter den 1998 vorgestellten mediaBlocks [50] von Ullmer, Ishii und Glas ist einen physischen Container für den Transport und die Möglichkeit der Manipulation von online verfügbaren digitalen Daten zu schaffen. Die Daten werden nicht in den mediaBlocks gespeichert, der mediaBlock enthält lediglich eine ID, die mittels eines RFID Readers ausgelesen werden kann, mit der die online bereitgestellten Daten verknüpft werden können. Dadurch ist der verfügbare Speicher defacto unbegrenzt, auch im Verlustfall eines mediaBlocks gehen keine Daten verloren.

Eine Möglichkeit der Benutzung ist der Transport von Daten. Hierzu sind an vier verschiedenen Geräten (Video Projektor, Netzwerk Drucker, Videokamera, digitales Whiteboard) Leseeinrichtungen angebracht. Sobald ein mediaBlock in diese Leseeinrichtung gesteckt wird kann die Aufzeichnung bzw. das Abspielen der darauf enthaltenden Daten beginnen. Eine weitere Möglichkeit ist, Daten von bzw. auf einen PC mittels einer herkömmlichen GUI-Anwendung zu transferieren.

Mithilfe des „media sequencers“ (sich Abbildung 7) wird es ermöglicht, den Inhalt verschiedener mediaBlocks zu manipulieren. Hierzu werden grundlegende Funktionen die einzelnen Sequenzen von mehreren mediaBlocks zu bearbeiten bereitgestellt und die fertig bearbeitete Sequenz auf einen neuen mediaBlock zu speichern. Die Manipulation der Daten erfolgt durch die Blöcke selbst.



Abbildung 7: Media sequencer für mediaBlocks [50]

Es wird leicht und verständlich gezeigt, wie einfach man aus einem normalen Holzblock ein TUI machen kann, wobei für einen tatsächlichen Einsatz solch eines Systems die Alltags-tauglichkeit noch aussteht. So ist immer eine Verbindung, mit relativ großer Bandbreite zum darunter liegenden Netzwerk notwendig, da auch Videodaten übertragen werden. Dies könnte spätestens beim Dreh des Urlaubsvideos mit der Videokamera zum Problem werden, zusätzlich wird durch das Senden der Daten der Stromverbrauch erhöht, was sich speziell bei mobilen Geräten auch negativ auswirkt.

Jeder Block repräsentiert Daten, man erhält darüber jedoch nur durch einen speziellen Browser nähere Informationen welche Daten es sind. Das ist im Vergleich zu den bisherigen Datenträgern keine Verschlechterung, jedoch wäre es möglicherweise hilfreich wenn an der Oberfläche des Blocks Informationen zum gespeicherten Inhalt angezeigt werden könnten, also beispielsweise das erste Bild eines Videos. Durch die Kopplung zwischen einer TUI und einer GUI-Anwendung mit einem Lesegerät am Computer kann wahrscheinlich die Akzeptanz der Benutzer gesteigert werden, da die gewohnte Kontrolle über die Daten nicht auf einem Schlag verloren geht.

2.7.3 ToolStone

Der 2000 von Rekimoto und Sciamarella entwickelte ToolStone (siehe Abbildung 8 und [43]) ist ein drahtloses Eingabegerät mit mehreren Freiheitsgraden welche die physikalische Manipulation am ToolStone selbst erkennt, diese Informationen können zur Steuerung von Anwendungen verwendet werden.

Die Bestimmung der Orientierung und der Position erfolgt mittels dreier Spulen die entsprechend im Quader angeordnet sind. Der ToolStone ist drahtlos, benötigt jedoch eine spezielle Unterlage, da es nur mit dieser möglich ist die Orientierung und Position zu erfassen. Zusätzlich muss ständig eine Kante oder eine Seite auf der Unterlage liegen um die Werte der Spulen erfassen zu können, der ToolStone kann somit nicht angehoben werden und ist auf einen fixen Arbeitsplatz beschränkt. Durch diese Einschränkungen kann man nicht von sechs vollständigen Freiheitsgraden sprechen.

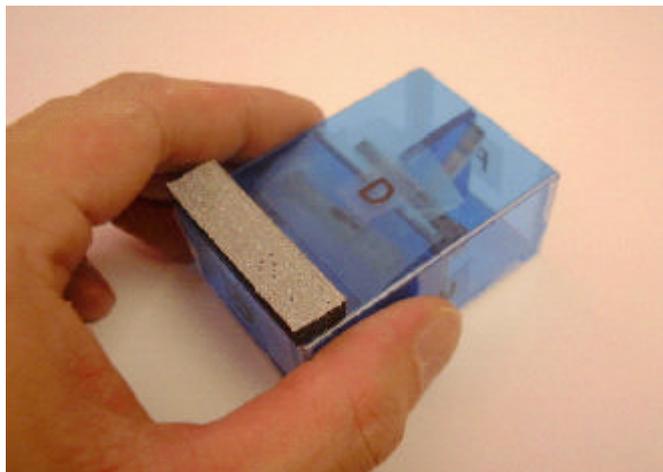


Abbildung 8: ToolStone [43]

Der ToolStone wurde speziell als Eingabegerät für Anwendungen konzipiert die mehr als zwei Freiheitsgrade, wie der Maus, benötigen oder bei denen die Bedienung mit mehr Freiheitsgraden vereinfacht wird.

Durch drehen, kippen usw. nimmt der ToolStone einen anderen Zustand ein, dieser kann erkannt werden und dient entsprechend als Eingabe für die Anwendung, zusätzlich kann der Benutzer mit einem Blick die Orientierung und Position erkennen und weiß somit auch in welchen Zustand sich der Stein gerade befindet. Hierzu wird weiters davon ausgegangen, das

speziell bei der Form eines Quaders, die Seiten entsprechend gekennzeichnet werden sollte, damit der Benutzer über die Auswirkung des aktuellen Zustandes jederzeit durch einen Blick Bescheid weiß. Bei einer Aufteilung mit 45° Abstand können 8 Richtungen unterschieden werden. Aufgrund der sechs Seiten ergeben sich insgesamt 48 Zustände die relativ schnell ausgewählt werden können.

Als Anwendungen, die für so ein Eingabegerät besonders geeignet sind, werden komplexere Szenarien angeführt, welche beispielsweise mehrere Schritte benötigen um eine Anweisung, wie das Zeichnen eines Kreises, auszuführen. Hierbei muss zuerst ausgewählt werden, dass ein Kreis gezeichnet werden soll, danach wird die Position festgelegt und der Radius muss auch noch angegeben werden.

Aufgrund der einfachen Auswechselbarkeit des ToolStones ist es möglich für unterschiedliche Benutzergruppen oder Anwendungen unterschiedliche Funktionalität zur Verfügung zu stellen, etwa einen, gegenüber Erwachsenen eingeschränkten oder vereinfachten Funktionsumfang für Kinder.

Natürlich ist auch eine beidhändige Bedienung möglich, also beispielsweise ein Zeichenstift in der einen Hand und der ToolStone in der anderen, weiters ist für sehr komplexe Anwendungen auch eine Kombination zweier ToolStones möglich.

Auch wenn die Interaktionsform mit dem Werkzeug eine Neue ist, so ist es den Testpersonen nach kurzer Zeit gelungen die grundlegenden Funktionen zu nutzen. [43]

Der ToolStone ist somit primär als zusätzliches Eingabegerät für PC-nahe Anwendungen gedacht, es sollen hiermit die bestehenden Eingabemöglichkeiten erweitert werden. Die Kontrolle selbst ist einfach, da man die Manipulation von Quadern bereits als Kleinkind lernt, jedoch nicht zwangsweise intuitiv. Die Bedienung basiert auf der Auswahl eines Zustandes, die dafür notwendige Bewegung muss jedoch in keiner Beziehung zur dazugehörigen Funktion stehen.

Die Einschränkungen die aufgrund der Technologie gegeben sind (anheben nicht möglich, durch die Unterlage vorgegebener Arbeitsbereich, etc.), können als akzeptabel eingestuft werden, da nur sehr spezielle Anwendungen als Zielgruppe angedacht sind.

2.7.4 Illuminating Clay

Mittels Illuminating Clay von Piper, Ratti und Ishii [40][41] lassen sich Landschaftsmodelle in Echtzeit analysieren und modellieren. In Abbildung 9 ist der Prototyp zu sehen an dem mehrere Personen gleichzeitig die Oberfläche modellieren. Normalerweise werden Landschaften als 3D-Modell mithilfe von Computern erstellt und mittels Virtual Reality dargestellt - oder es werden physikalische Modelle nach einem bereits bestehenden Plan angefertigt. Der Umgang mit virtuellen 3D-Modellen ist oft nicht sehr einfach, da die Bedienung komplex ist und die Zusammenhänge zwischen der Bedienung, den dargestellten Änderungen und den daraus resultierenden Auswirkungen nur schwer zu erkennen sind. Illuminating Clay soll bei den Schwachstellen konventioneller Systeme Lösungen anbieten indem die Modellierung mittels eines TUI erfolgt und zusätzlich weitere Informationen oder bereits Resultate eingeblendet werden. Die Modellierung der Landschaft erfolgt mit einer Knetmasse die auf einem Metallgeflecht aufgebracht ist, somit kann die Form der Oberfläche einfach mit den Fingern verändert werden. Die Höheninformationen werden mithilfe eines Laser Scanners, der über der Arbeitsfläche angebracht ist, ermittelt. Durch einen Projektor können direkt auf die Knetmasse zusätzliche Informationen eingeblendet werden, dies reicht von einfachen Höhenlinien, farblichen Kennzeichnungen von Flächen die bestimmte Eigenschaften aufweisen bis zum Darstellen von fließenden Gewässern. Bei den fließenden Gewässern ist es beispielsweise möglich, die Auswirkungen der Modellierung in Echtzeit

darzustellen, das bedeutet dass die Resultate der Änderungen an der Knetmasse sofort ersichtlicht werden und somit auch komplexe, nicht intuitive und dynamische Zusammenhänge einfach dargestellt werden können.

Der Vorteil liegt eindeutig in der Art der Modellierung, da hierbei die Fähigkeiten des Menschen im Umgang mit physikalischen Objekten ausgenutzt werden, und der Benutzer eine unmittelbare Rückmeldung über die Auswirkungen seines Handels bekommt.



Abbildung 9: Illuminating Clay im Einsatz [40]

2.7.5 Kontextbestimmung mittels Accelerometer

Es existieren einige sehr ähnliche Ansätze (z.B. [16], [28] und [51]) zur Ermittlung von Kontextinformation mithilfe von Beschleunigungssensoren. Meistens werden mehrere Sensoren an speziellen Stellen am Körper, wie etwa an einem Knie, an der Hüfte oder an den Armen, angebracht. Diese Sensoren liefern über einen längeren Zeitraum Daten mit denen versucht wird eine Aussage über die aktuelle Tätigkeit des Trägers der Sensoren treffen zu können. Da die Werte bei oftmaliger Wiederholung immer wieder sehr unterschiedlich ausfallen können und sich auch unterschiedliche Bedingungen in der Umgebung (unterschiedliche Stiegen, unebener Boden, unterschiedliches Mobiliar, etc.) auswirken, wird versucht die aktuelle Tätigkeit unter Zuhilfenahme von statistischen Verfahren mit möglichst großer Wahrscheinlichkeit zu erkennen. Die Zustände die erkannt werden können sind jedoch noch relativ stark eingeschränkt, so handelt es sich in Beispielen beim Stiegensteigen immer um eine bestimmte Stiege oder der Sessel und die Haltung beim Sitzen auf diesem Sessel ist festgelegt. Auch die Anzahl der unterschiedlichen Zustände die erkannt werden können, ist teilweise eher gering. Je nachdem wo und wie viele Sensoren verwendet werden lassen sich beispielsweise gehen, sitzen, laufen, Stiegen steigen (hinauf und hinunter) und Rad fahren erkennen. Es werden also keine einzelnen Bewegungen erkannt sondern immer nur längerfristige Tätigkeiten.

Bei den Anwendungen werden meistens preiswerte Sensoren mit lediglich zwei Freiheitsgraden oder einem eingeschränkten Wertebereich eingesetzt. Die Einschränkung der Freiheitsgrade ist meist kein Problem, da die Freiheitsgrade der Gelenke an denen die Sensoren befestigt sind auch eingeschränkt sind.

Aufgrund der erforderlichen Positionierung der Sensoren ist es nahe liegend, dass diese Ansätze im Bereich des Wearable Computing umgesetzt werden können, hierbei muss die Erkennungsrate jedoch noch gesteigert werden und die Recheneinheiten die für die Auswertung der Sensordaten verantwortlich sind müssen auch kleiner bzw. sogar in die Kleidung integriert werden. Ein längerfristiger autonomer Betrieb muss auf jeden Fall gewährleistet werden können. Durch die Kopplung mit anderen Sensoren kann eine weitere Steigerung der Erkennungsrate erreicht werden. Hierbei könnte es sich etwa um einen GPS

Empfänger oder auch um Lichtsensoren handeln. Mit deren Hilfe kann einfacher entschieden werden, ob sich der Benutzer in einem Gebäude oder im Freien aufhält - dadurch kann die Kontexterkenkung verbessert werden.

2.7.6 BlueWand

Der BlueWand IIX (siehe [1] und [25]) der Firma BeeCon bietet vergleichbare Eigenschaften wie der im Rahmen dieser Arbeit entwickelte Hardwareteil. Der BlueWand verfügt über insgesamt sieben Bewegungssensoren, drei Gyroskope, die die Drehbeschleunigung für jeweils eine Raumachse erfassen und vier Accelerometer, die für jede Raumachse die lineare Beschleunigung messen. Zusätzlich stehen noch drei Taster als Eingabemöglichkeit zur Verfügung. Aufgrund der Sensorik ist es laut Hersteller möglich, jede beliebige Bewegung im Raum zu erfassen. Bei näherer Betrachtung der Spezifikation relativiert sich diese Aussage jedoch wieder. Die Gyroskope können Drehungen bis zu $300^\circ/s$ erfassen. Bei Messungen mit der selbst entwickelten Hardware wurden bei schnellen Handbewegungen jedoch Drehungen mit $900^\circ/s$ und mehr gemessen. Dies kann auch ohne Sensorik überprüft werden, indem man beispielsweise die Handfläche in einer bestimmten Zeit mehrmals um 180° hin und her bewegt. Daraus ist ersichtlich, dass die Erkennung von sehr schnellen Bewegungen nur mit Einschränkungen möglich ist.

Die Übertragungsrate von 50Hz mag für die Bewegungserkennung ausreichen, für andere Anwendungen könnte die daraus resultierende Latenz von bis zu 20 Millisekunden alleine durch die Übertragung und zusätzlich die Dauer der eigentlichen Messung. Weitere Angaben über wesentliche Parameter wie die Genauigkeit, die Wertebereiche, Störanfälligkeit bei sehr langsamen Bewegungen, Drifts einzelner Werte, Auswirkungen von Störgrößen wie Temperatur, starke Magnetfelder usw. werden von Seiten des Herstellers jedoch nicht gemacht. Das Hauptaugenmerk liegt in der Kommunikation und dem verwendeten Bluetooth-Stack und nicht in der Sensorik.

Die Datenübertragung erfolgt mittels Bluetooth, dadurch ist ein universeller Einsatz des BlueWand IIX gewährleistet. Die übertragenen Werte sind Rohwerte der Sensoren die den gemessenen Spannungen und Puls-Pausen-Verhältnissen entsprechen, die Umrechnung dieser Daten kann mit der mitgelieferten Software erfolgen. Im Lieferumfang ist eine Maustreiberemulation, jedoch keine Software für die Bewegungserkennung mit einem benutzerfreundlichen API.

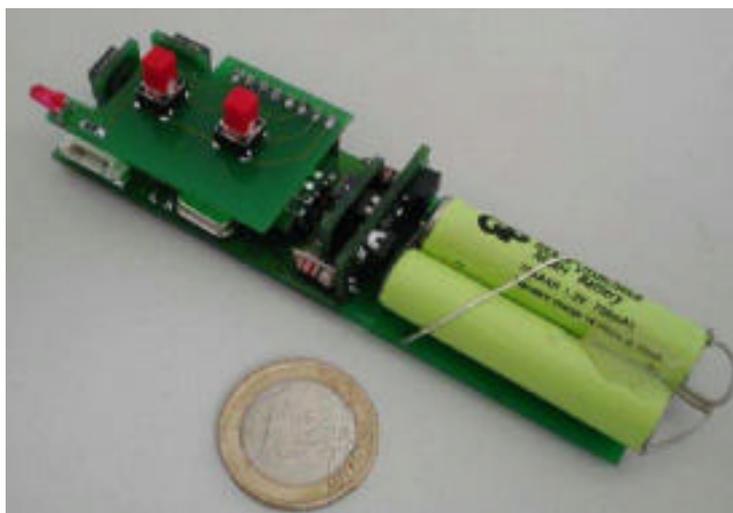


Abbildung 10: BlueWand IIX [1]

2.7.7 Würfel als TUI

In Folge werden drei verschiedene Würfel vorgestellt die als TUI dienen und der vorliegenden Arbeit vom Grundgedanken her sehr ähnlich sind. Es handelt sich jeweils um einen Würfel der mittels integrierter Sensorik Daten über die Bewegung aufzeichnen kann. Diese, teilweise bereits vor verarbeiteten, Daten werden anschließend mittels einer drahtlosen Datenübertragung weitergegeben. Aufgrund der Aktualität dieser Arbeiten – zwei sind in den letzten zwölf Monaten erschienen – ist ersichtlich, dass dieses Gebiet für die Forschung aktuell sehr interessant ist.

Bewegungserkennung mittels Trägheitssensoren

Mithilfe einer „inertial measurement unit“ (IMU, Abbildung 11) in Würfelform versuchten Benbasat und Paradiso [2] bereits 2002 Trägheitssensordaten aufzuzeichnen um daraus Bewegungen zu erkennen. Die Sensorik der IMU ermöglicht es Änderungen von sechs Freiheitsgraden zu erkennen, somit kann jede Bewegung im 3D-Raum erkannt werden. Aufgrund der preisgünstigen und kleinen Sensoren (Seitenlänge des Würfels ist rund 3,2cm) ist es jedoch nicht möglich die Orientierung länger als fünf Sekunden zu beobachten. Die Datenübertragung erfolgt mittels proprietärer Funktechnologie und erfordert beim Empfänger eine entsprechende Hardware.

Die Gesten, die erkannt werden können, werden aus atomaren Gesten zusammengesetzt, diese Zusammenstellung liegt im Aufgabenbereich des Anwendungsentwicklers, wobei es hier bei der Erkennung auch Einschränkungen bei mehrdimensionalen Bewegungen gibt. Auch bei der Betrachtung von Gesten innerhalb einer kontinuierlichen Bewegung kann aufgrund des Auswerteverfahrens die Geste nicht immer erkannt werden.

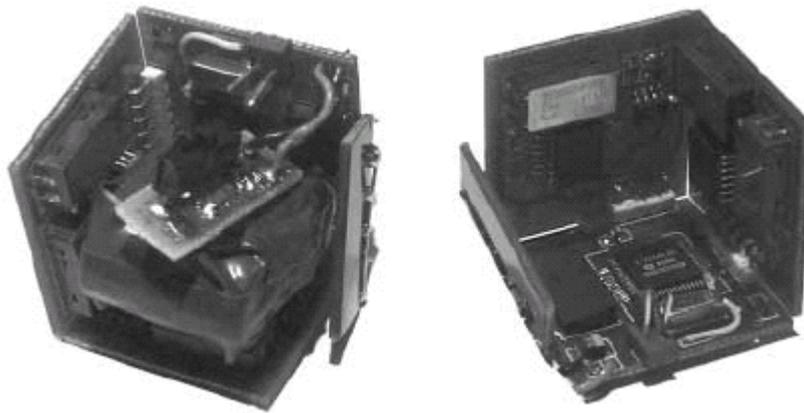


Abbildung 11: IMU Hardware [2]

Autonomer Würfel zur grundlegenden Navigation und Eingabe

Ende 2003 wurde ein weiterer Würfel vorgestellt mit dem es möglich ist, einfache Navigations- und Eingabeaufgaben durchzuführen (siehe [52], Abbildung 12). Die Sensorik besteht aus insgesamt drei Sensoren. Zwei Beschleunigungssensoren für die Bestimmung der Orientierung. Beim dritten Sensor handelt es sich um einen kapazitiven Sensor mit dem der Mikrocontroller aus dem Standby-Zustand aufgeweckt wird, sobald der Würfel berührt wird. Durch eine Datenübertragung mittels RF-Funkmodul und dem Einsatz einer Batterie kann der Würfel autonom betrieben werden.

Die Auswertung der Bewegungen erfolgt bereits mit einem Mikrocontroller im Würfel, d.h. es werden nur die bereits erkannten Bewegungen übermittelt. Das hat zur Folge, dass die Algorithmen einfach gehalten werden müssen und es Einschränkungen bei der Erkennung komplexerer Gesten gibt. Die Zustände des Würfels, die erkannt werden können, sind relativ

stark eingeschränkt. Es ist etwa die Bestimmung der Seite des Würfels welche oben liegt oder die Orientierung der Würfelflächen (Granularität von 90°) relativ zum Benutzer, wobei hier die Startposition bekannt sein muss, möglich. Weiters können noch drei einfache Bewegungen erkannt werden, diese sind schütteln, drehen und das Klopfen auf den Würfel.

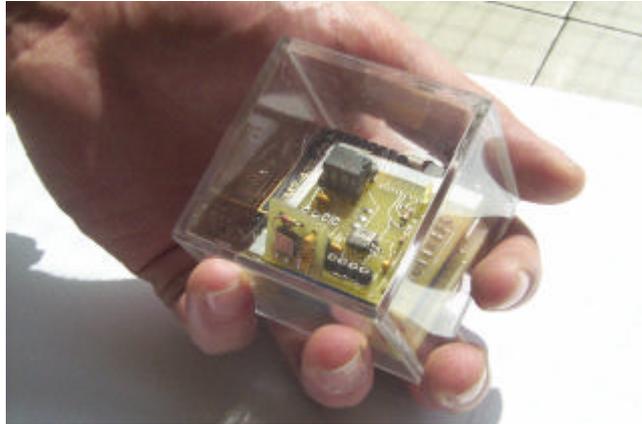


Abbildung 12: Hardware in transparenten Kunststoffwürfel [52]

iCube

Pering [38] stellt 2004 eine weitere Variation der Würfel TUIs, den iCube (Abbildung 13), vor. Neben den nicht näher erläuterten Accelerometern, die für die Bewegungserkennung zuständig sind, verfügt der iCube auch noch über einen RFID-Tag. Die Sensordaten werden mittels Bluetooth gesendet.

Als Anwendungsszenarien werden Beispiele aus dem öffentlichen Raum angeführt. So soll mit einem Handscanner die ID des Würfels gelesen werden, daraufhin verbindet sich eine kleine mobile Recheneinheit, welche man ebenfalls bei sich trägt, mittels Bluetooth mit dem iCube. Aufgrund der ID soll auch ein Verbindungsaufbau, z.B. zu einer öffentlichen Anzeige, für die der iCube als Bedienelement gedacht ist, möglich sein.

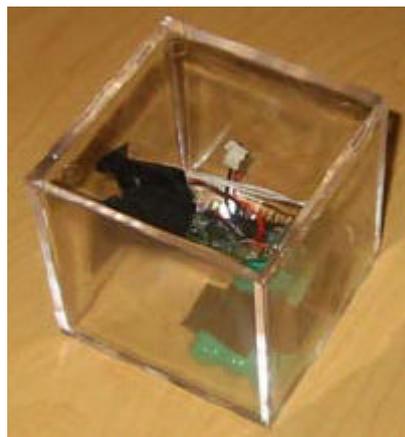


Abbildung 13: Prototyp des iCube [38]

Über die Qualität, den Umfang und die Möglichkeiten des Prototyps bei der Bewegungserkennung kann aufgrund der sehr spärlichen Angaben über die Hard- und Software keine Aussage gemacht werden, es wird jedoch mehrfach erwähnt, dass der iCube als Eingabegerät für einfache Anwendungen, mit entsprechend leicht zu erkennenden Gesten, gedacht ist.

2.8 *TangibleControl*

TangibleControl, also der im Rahmen dieser Arbeit entwickelte Prototyp, schließt thematisch beinahe nahtlos an die Beispiele in 2.7.7 Würfel als TUI an, auch wenn es auf Teilbereichen wesentliche technische Unterschiede gibt. Konzeptionell gesehen liegt, im Gegensatz zu den angeführten Anwendungen, das Hauptaugenmerk nicht am Hardware- bzw. Elektronikbereich, sondern auf der Gestaltung des Frameworks mit den entsprechenden, einfach zu nutzenden, Möglichkeiten.

Um bei der Gestenerkennung nicht durch die Hardware eingeschränkt zu sein, wird bei der Wahl der Sensorik stark nach der Aktualisierungsrate, Genauigkeit sowie dem Wertebereich bewertet, der Preis hingegen spielt eine untergeordnete Rolle. Auch bei der Datenübertragung sind die technischen Merkmale, wie auch die Standardisierung, im Vordergrund. Durch diese Bewertung ergeben sich im Vergleich zu den bereits existierenden Arbeiten auch geringere Einschränkungen die durch die Hardware bedingt sind und somit eine größere Anzahl von Gesten die maximal erkannt werden können. Dies erfordert auch, dass die Sensorwerte über einen längeren Zeitraum stabil erkannt werden können.

Da die Sensordaten an eine Auswertestation übertragen werden und nicht direkt durch das TUI verarbeitet werden gibt es fast keine Einschränkungen beim Umfang der Algorithmen für die Bewegungserkennung. Weiters ist der Ansatz bei TangibleControl nicht auf die Form eines Würfels eingeschränkt sondern es kann durch die Software eine Anpassung an beliebige Formen erfolgen.

Die größten Unterschiede ergeben sich durch die Software bzw. durch den Ansatz der dahinter steckt. Bei der Software handelt es sich um ein einfach zu verwendendes API das bereits eine Vielzahl von unterschiedlichen Gesten für die verschiedenen geometrischen Formen zur Verfügung stellt, und nicht nur um eine Software zur Auswertung der Sensordaten. Dies erlaubt auch die Anwendung der Gestenerkennung ohne Wissen über die Hardware und die Sensorik.

Aus der Gestaltung des APIs resultiert die freie Abbildung der erkannten Gesten auf Aktionen der Anwendungsprogramme.

3 Gestenerkennung

Das Ziel, die Kommunikation mit der Maschine möglichst intuitiv zu gestalten, erfordert es, das menschliche Verhalten in den Kommunikationsprozess mit einzubeziehen. Die Gestik, welche ein Teil des menschlichen Kommunikationssystems ist, soll somit auch als Eingabemöglichkeit zur Verfügung stehen. Hierzu ist es notwendig einige wichtige Eigenschaften der Gestik zu berücksichtigen:

- Gestik ist weit verbreitet und Bestandteil des täglichen Lebens.
- Trotz fehlender bzw. nicht möglicher Standardisierung existiert ein allgemeines Verständnis, welche Bedeutung Gesten haben.
- Gegenstände und Personen aus der Umwelt werden in Gesten miteinbezogen.
- Gesten können in verschiedenen Kulturen unterschiedliche Bedeutungen besitzen, auch lokale Unterschiede existieren, sie sind jedoch nicht an sprachliche Grenzen gebunden.
- Gesten werden oft unterbewusst eingesetzt und sind gegebenenfalls auch leicht zu erlernen.
- Gesten werden verwendet um andere Kommunikationskanäle (z.B. Sprache) zu unterstützen oder zu begleiten.
- Gestik kann als einzige, isolierte Eingabe dienen.

3.1 Begriffsbestimmung

In der Literatur findet man eine Vielzahl an unterschiedlichen Definitionen des Begriffes Geste. Dies ist darauf zurückzuführen das Gesten in einer großen Anzahl von Anwendungen verwendet werden und dies zieht somit auch jeweils eine unterschiedliche, auf die Anwendung bezogene, Definition mit sich. Die Definition die in diesem Dokument verwendet wird lautet:

Geste: Eine ausdrucksvolle Bewegung oder Pose des Körpers, im Besonderen durch die Hände, um gewöhnliche oder spezielle Signale zu übermitteln.

Freeman und Roth [13] unterscheiden zwei Kategorien von Gesten:

- statisch
- dynamisch

Eine statische Geste ist eine einzelne Pose oder ein bestimmter Zustand des Körpers oder der Hand und kann somit durch ein einziges Bild dargestellt werden. Die Information wie es zu diesem Zustand gekommen ist, ist also nicht erforderlich. Bei einer dynamischen Geste hingegen handelt es sich um eine bewegte Geste, welche durch eine zeitliche Abfolge von statischen Gesten definiert wird. Daraus ergibt sich, dass dynamische Gesten eine räumliche Ausdehnung besitzen. Es bewegen sich also beispielsweise die Hand oder der Kopf im Raum. Durch die Dauer der Bewegung und den räumlichen Eigenschaften ergeben sich zusätzliche Parameter, wie etwa Geschwindigkeit, Zeit- und Ortsveränderung, welche eine dynamische Geste beschreiben können.

Als Beispiel für eine dynamische Geste kann das zustimmende Nicken mit dem Kopf genannt werden. Hier ist eine zusammenhängende Folge von einzelnen Posen notwendig um als Nicken erkannt zu werden. Ein Bild, mit dem auf einer Hand aufgestützte Kopf, mit einem

gelangweilten Gesichtsausdruck könnte hingegen ein Beispiel einer statischen Geste für Langeweile sein.

Im Weiteren wird der Begriff *Geste* oder allgemeiner *Bewegung* sowohl für statische als auch für dynamische Gesten verwendet. Wenn eine Unterscheidung zu dynamischen Gesten notwendig ist, so werden die Begriffe "Pose", "statische Geste" oder im softwarenahen Teil auch "Zustand" verwendet.

Normalerweise treten statische und dynamische Gesten nacheinander auf. Oft liegt keine zusätzliche Information vor, welche eine eindeutige Trennung der Sequenz, in für Gesten relevante und irrelevante Teile erlaubt. Auch die Trennung der relevanten Daten für einzelne, nacheinander ausgeführte Gesten ist mitunter schwierig. Dies ist als weiteres Problem in der Gestenerkennung zu sehen, wobei der Begriff „Gestenerkennung“ mit folgender Definition verwendet wird:

Gestenerkennung: Das Problem, Gesten in einer gegebenen Menge von Eingangsdaten zu erkennen, wobei die Eingangsdaten eine oder mehrere Gesten enthalten können, jedoch nicht müssen.

Die Gestenerkennung wird weiters erschwert, da es kaum gelingt eine Geste mehrmals nacheinander völlig identisch durchzuführen. Dies hat zur Folge, dass die Eingangsdaten aus denen die Gesten erkannt werden sollen unterschiedlich sind. Weiters haben die Eingangsdaten (meist Sensordaten) oft eine viel größere Auflösung als durch den Benutzer beeinflussbar, umfangreicher als erforderlich und somit sind nicht alle Teile der Information relevant, mit Ausreißern behaftet oder hängen zusätzlich von anderen Größen ab die vom Benutzer nicht kontrollierbar sind. Somit ist eine eindeutige Erkennung oft nur schwer oder gar nicht möglich, es wird folglich versucht die Eingangsdaten mit einer möglichst hohen Wahrscheinlichkeit einer Geste zuzuordnen. Dies ist natürlich ein Nachteil zu herkömmlichen Eingabemöglichkeiten wie etwa die Taste einer Maus, wobei hier die Erkennung der Geste bereits durch die Hardware erfolgt. Es muss ein bestimmter Druck an einer bestimmten Stelle überschritten werden um als Klick erkannt zu werden. Diese Auswertung erfolgt durch den entsprechenden Schalter der diese Information dann weiterleitet. Aufgrund der Art des Schalters ist es unerheblich ob etwas, oder viel stärker gedrückt wurde als durch den Druckpunkt der Maustaste erforderlich ist. Die unterschiedliche physikalische Größe (Stärke des Tastendrucks) wird somit bereits durch den Sensor, welcher in diesen Fall ein einfacher binärer Schalter ist, in zwei disjunkte Mengen geteilt.

Zusätzlich werden individuelle Unterschiede verschiedener Benutzer durch die Hardware der Maus ausgeglichen. Es ist folglich nicht erforderlich, die Maus für jeden Benutzer extra einzustellen bzw. zu kalibrieren. Dies kann man aus Benutzersicht jedoch auch als Nachteil werten, da sich der Benutzer an das Eingabegerät anpassen muss, indem er den vorgegebenen Druckpunkt der Maustaste überwindet und nicht mit einem frei definierbaren Druck das Drücken der Maustaste auslösen kann, und nicht umgekehrt.

3.2 Anforderungen an Gestenerkennungssysteme

Unter anderen von Brauer [4] werden neben den für Software allgemeinen Qualitätskriterien wie Korrektheit, Effizienz, etc. noch weitere Anforderungen an ein Gestenerkennungssystem gestellt:

- Um die Gestenerkennung möglichst universell und somit für eine große Anzahl von unterschiedlichen Anwendungen und Szenarien zu gestalten sollten die Gestiken *frei definierbar* sein, außer es besteht ein ausreichender Vorrat an statisch vorgegebenen

Gesten der eine umfassende Kommunikation trotz fehlender Standardisierung der Gesten ermöglicht.

- Wenn es aufgrund der eigentlichen Anwendung nicht möglich ist die Erkennung der Gesten zu testen, muss ein eigenes *Trainingswerkzeug* vorhanden sein.
- Wenn die Gestik zur Kommunikation mit anderen Systemen verwendet wird muss die *Erkennung ohne signifikante Zeitverzögerung* erfolgen, um bei Bedarf eine prompte Rückkopplung zu ermöglichen.
- Um die Akzeptanz zu gewährleisten und eine problemlose Kommunikation zu ermöglichen muss die *Erkennungsrate möglichst hoch* sein. Werte nahe 100% sind wünschenswert, speziell wenn die durchgeführte Geste weitreichende Auswirkungen auf die Anwendung bzw. den gesamten Kommunikationsverlauf hat. Des Weiteren kann es sinnvoll sein das Risiko fehlerhafter Klassifizierung bzw. Erkennung zu minimieren, indem Grenzfälle als nicht erkennbar zurückgewiesen werden. Dieses Verhalten des Erkennungssystems sollte jedoch steuerbar sein um eine optimale Anpassung an die Anwendung zu ermöglichen.
- Da die Gestenerkennung nur als Mittel zur Kommunikation verwendet wird und nicht als die eigentliche Anwendung gesehen werden kann, muss die Gestenerkennung in einem übergeordneten System *einfach zu integrieren* bzw. zu verwenden sein. Dem Anwendungsprogramm müssen die Informationen über die ausgeführten Gesten zur Verfügung gestellt werden. Auch einzelne Parameter der Gesten, nicht bereits auf Gesten-Level abstrahiert, sollen für anwendungsspezifische Funktionen bereitgestellt werden.

3.3 Vorgehen bei der Gestenerkennung

Abhängig vom Gestenerkennungssystem sind verschiedene Arbeitsschritte bis zur Erkennung der Gesten erforderlich, diese können bei den einzelnen realisierten Systemen im Umfang und in den daran gestellten Anforderungen stark variieren. In Abbildung 14 ist eine grobe Einteilung in die erforderlichen Schritte zu sehen die in den meisten Systemen vorhanden sind.



Abbildung 14: Vorgehensweise bei der Gestenerkennung

Im ersten Schritt erfolgt die Aufnahme der Daten. Dieser Schritt umfasst den Bereich der Sensorik, dies kann beispielsweise eine Kamera, ein Orientierungssensor oder auch ein Datenhandschuh sein. Es werden die Rohdaten ohne spezielle Aufbereitung für die Gestenerkennung direkt vom Sensor übernommen.

In einer ersten Vorverarbeitung werden die relevanten Informationen aus den Rohdaten gefiltert. Hierbei wird etwa in einem Bild die zu verfolgende Hand gesucht und die Umgebung bzw. der Hintergrund wird ignoriert, da dies für die Bewegungserkennung irrelevante Informationen sind.

Im nächsten Schritt werden die erforderlichen Merkmale berechnet. Dies kann eine einfache Normalisierung der Werte sein, die Berechnung der Silhouette oder etwa die Bestimmung der Konturen der Hand in einem Bild.

Bei der Klassifizierung werden in vielen Anwendungen einfache Endliche Automaten, Neuronale Netze oder Hidden-Markov-Modelle verwendet. Hierbei wird meistens versucht die Geste aus einem Gestenvorrat zu bestimmen die am Besten mit der durchgeführten Geste übereinstimmt.

Als Resultat wird in der überwiegenden Mehrzahl der Verfahren genau eine oder keine Geste aus einem bekannten Gestenvorrat geliefert. Nur in wenigen Fällen werden allgemeine Parameter über die durchgeführte Geste zur Verfügung gestellt.

Diese allgemeine Vorgehensweise hängt natürlich von der genauen Anwendung ab. So ergeben sich erstens aufgrund der unterschiedlichen Sensoren verschiedene Anforderungen an die Auswertung und zweitens spielt es auch eine Rolle von welchem (Körper)Teil die Geste durchgeführt wird. Die Verfahren lassen sich auch nach eben diesen (Körper)Teilen deren Bewegungen verfolgt werden klassifizieren. Der Großteil der Verfahren zur Gestenerkennung kann zumindest einer der folgenden vier Kategorien zugeteilt werden, wobei sich teilweise eine Überlappung bei der Zuordnung ergeben kann:

- Handgesten
- Körpergesten
- Erkennung von Zeichensprache
- Gesten von Artefakten (Stifte, Würfel, Mäuse und dergleichen)

Eine Trennung zwischen Hand- und Körpergesten erfolgt aufgrund der großen Anzahl der Systeme die Handgesten verwenden. In dieser Einteilung werden auch Gesten die mit einzelnen Fingern einer Hand durchgeführt werden zu den Handgesten gezählt, die größte Einheit ist jedoch die Hand. In den Bereich der Körpergesten fallen somit Bewegungen die von den Armen oder beispielsweise auch vom Kopf bzw. der Nase durchgeführt werden. Weiters werden auch längerfristige Gesten wie gehen, Stiegen steigen, sitzen usw. der Gruppe der Körpergesten zugeordnet, auch wenn diese auf den ersten Blick nicht als Gesten zu erkennen sind. Natürlich wären bei den Körpergesten noch weitere Unterteilungen in kleinere Gruppen wie Arm, Augen, Kopf, Oberkörper usw. möglich und bei einer detaillierteren Betrachtung durchaus angebracht und sinnvoll.

Auch die Erkennung der Zeichensprache fällt großteils in den Bereich der Körpergesten, es wurde jedoch trotzdem als eigener Bereich gewählt, da diese Anwendungen verhältnismäßig gut isoliert betrachtet werden können, da einerseits die Gesten, bis auf Unterschiede in den einzelnen Zeichensprachen, ähnlich sind und die Anwendung ist die Gleiche und steckt bereits indirekt in der Kategorie. Die Erkennung der Zeichensprache ist einerseits für Leute mit einer Behinderung interessant, andererseits können mit diesen Gesten auch beliebige andere Anwendungen gesteuert werden. Diese profitieren dadurch von einem sehr umfangreichen und bekannten Gestenvorrat.

Die bisherigen Gruppen sind sehr stark von den traditionellen Anwendungen gefüllt, dies trifft bei der Bewegungserkennung von Artefakten nicht so stark zu. Hierbei spielt es keine Rolle welche Bewegung etwa eine Hand vollzieht, sondern es ist nur der Gegenstand von Interesse der durch diese Hand bewegt wird. In vielen Fällen sind diese Bewegungen zwar identisch, dies muss jedoch nicht so sein. Beispielsweise kann ein Gegenstand in der Hand gedreht werden, oder er wird festgehalten und die Hand mit dem Gegenstand wird gedreht. Ein Rückschluss von der Bewegung des Artefaktes auf eine Geste des Auslösers dieser Bewegung ist somit nur sehr eingeschränkt möglich. In die Gruppe „Gesten von Artefakten“ fallen auch diese und ähnliche Arbeiten wie etwa [1], [38] und [52]. Die prinzipielle Vorgehensweise zur Bewegungserkennung ist identisch mit den anderen Kategorien, jedoch sind nicht alle bisherigen Werkzeuge verwendbar, so ist etwa ein Datenhandschuh für diese Anwendung nicht geeignet, für Handgesten kann er jedoch erste Wahl sein.

Da in der Gruppe „Gesten von Artefakten“ ein eigenes Artefakt verwendet wird, welches manipuliert wird, ist es nahe liegend diesen Gegenstand mit zusätzlicher Sensorik zu versehen. Dies entspricht wieder dem Ansatz des Datenhandschuhes, hier wird auch versucht die Hand mit zusätzlichen Sensoren auszustatten, auch wenn dies wesentlich unnatürlicher ist, als

es in dieser Gruppe der Fall sein kann. Diese Eigenschaft, dass die Sensorik in diese Artefakte ohne Beeinflussung dieser integriert werden kann, kann entsprechend ausgenutzt werden.

3.4 Vergleich der Verfahren zur Gestenerkennung

Die Arten der Sensoren die hauptsächlich für die Gestenerkennung verwendet werden, können in drei Gruppen eingeteilt werden. Die Gruppe der Datenhandschuhe ist, wie der Name schon sagt, auf die Erkennung von Handgesten beschränkt. Bei der zweiten Gruppe, den Computer-Vision-basierten Systemen, ist diese Einschränkung nicht gegeben - diese Verfahren sind somit universeller einsetzbar. Die dritte Gruppe umfasst die Erkennung von Bewegungen von Artefakten die mit einer eigenen Sensorik für die Bewegungserkennung ausgestattet sind. Die Vor- und Nachteile der jeweiligen Ansätze sind in Tabelle 1 gegenübergestellt.

3.5 Erkennung mittels Datenhandschuh

Niethard und Pfeil [36] zeigen die Beweglichkeit der Hand und der Arme. Daraus lässt sich erkennen welche vielfältigen Möglichkeiten zur Gestendarstellung alleine durch eine Hand möglich sind.

Sturman [49] etwa kommt bei seiner Zählung der Freiheitsgrade einer Hand auf 23. Hierbei ist das Handgelenk (2 Freiheitsgrade) bzw. die Positionierung der Hand (3 Freiheitsgrade) noch nicht berücksichtigt. Teilweise bestehen Abhängigkeiten und auch der Bewegungsspielraum der einzelnen Gelenke ist unterschiedlich, trotzdem gehört die Hand zu den flexibelsten Körperteilen die in der Natur vorkommen. Somit ist ersichtlich das alleine durch die Betrachtung einer Hand eine große Menge an Informationen zur Verfügung stehen welche z.B. zur Gestenerkennung verwendet werden können.

Auf einem Handschuh sind Sensoren angebracht durch die es möglich ist, Informationen über die Position der Finger zu erhalten. Die Messung der physikalischen Größen (Neigung der Finger) aktueller Datenhandschuhe basieren auf der Messung mittels optischen Methoden oder der Widerstandsmessung.

3.5.1 Entwicklung der Datenhandschuhe

Bereits 1976 wurde an der Universität von Illinois zum ersten Mal (siehe [48][53]) die Idee eines Datenhandschuhes, dem so genannten Sayre Glove (siehe Abbildung 15), entwickelt. Entlang der Finger des Handschuhes wurden dünne, flexible, lichtdurchlässige Röhren befestigt. An einem Ende der Röhre wurde eine Lichtquelle, am anderen Ende eine Fozelle angebracht, welche in Abhängigkeit der einfallenden Lichtmenge den Widerstand veränderte. Aufgrund der daraus resultierenden Spannung an den Fozellen konnte die Krümmung der Finger abgeschätzt werden. Der Sayre Glove wurde nicht zur Gestenerkennung sondern zur multidimensionalen Steuerung, z.B. von Schieberegeln, verwendet.

Der erste Datenhandschuh der zur Gestenerkennung bestimmt war, wurde 1983 von Grimes mit dem Digital Data Entry Glove realisiert. Von VPL Research wurden 1987 die ersten kommerzielle Datenhandschuhe, der DataGlove (siehe Abbildung 15) und der Z-Glove, vorgestellt [27]. Beide Handschuhe wurden als universelle Eingabemöglichkeit etwa für die Manipulation von Objekten durch die Hand, zur Navigation in der virtuellen Realität oder zur Eingabe der amerikanischen Gebärdensprache (american sign language, ASL) entworfen. Die Messung erfolgt mittels Goniometer (Winkelmesser) welche von Zimmerman 1985 patentiert wurde. Hierbei werden optische Fasern zur Bestimmung des Beugegrades der Finger eingesetzt. Zusätzlich zur relativen Position der Finger konnten noch die Orientierung und die Position der Hand bestimmt werden. Hier liegt auch der Unterschied der beiden Handschuhe.

	Datenhandschuh	Computer Vision	Sensorik in Artefakten
Kosten der Hardware	Hohe Kosten für präzise Geräte.	Geringe Anschaffungskosten	Geringe Anschaffungskosten.
Natürlichkeit und Akzeptanz	Der Handschuh wird oft als beengend empfunden – dies beeinträchtigt die Natürlichkeit der Bewegung und des Umgangs mit dem Handschuh. Das Anlegen ist lästig, hinzukommen hygienische Aspekte beim tragen durch mehrere Personen (Schwitzen der Hand). Teilweise vorhandene Kabel schränken die Bewegungsfreiheit des Benutzers ein.	Computer Vision entspricht dem Sinn mit dem Menschen den Großteil ihrer Informationen über ihre Umgebung aufnehmen, die Methode ist daher natürlich. Die Erkennung erfolgt berührungsfrei und für den Nutzer unbewusst. Diese Methoden erzielen daher eine hohe Akzeptanz.	Die Artefakte werden nur geringfügig verändert (Gewicht), Idealerweise ändert sich die Größe nicht, somit ergibt sich auch keine Verschlechterung der Akzeptanz. Eigens entwickelte Gegenstände können auf die Anforderungen der Benutzer und die Anwendungen angepasst werden.
Handgröße und -anatomie	Probleme bei großen oder kleinen Händen. Bei ungenauer Passform kann es zusätzlich zu ungenauen Messwerten führen und somit die Erkennungsgenauigkeit reduzieren.	Es sind keine daraus resultierende Probleme zu erwarten.	Die Größe der Artefakte ändert sich nur gering oder gar nicht, somit tritt keine Verschlechterung der Anatomie ein. Die Handgröße spielt keine Rolle, jedoch sollen von der Größe her angemessene Artefakte verwendet werden.
Umgebungsunabhängigkeit	Die Bestimmung der Handparameter ist unabhängig von der Beleuchtung, dem Hintergrund und der Sichtbarkeit der Hand.	Die Verfahren sind abhängig von der Beleuchtung und zum Teil auch vom Hintergrund. Bei Sichtverdeckungen der Hand können keine Aussagen gemacht werden.	Keine Abhängigkeit von der Umgebung, da nur die Daten über das Artefakt aufgezeichnet werden.
Genauigkeit, Geschwindigkeit	Die Bestimmung der Handparameter ist präzise und sehr schnell (bis zu 300 Hz). Die Genauigkeit von 3D-Magnet-Sensoren zur Positionsbestimmung nimmt mit dem Abstand zwischen Sensor und Empfänger ab. Ferromagnetische Objekte im Raum verzerren das Ergebnis.	Aussagekräftige Merkmale sind schwierig zu gewinnen, die Verfahren sind oft kompliziert und benötigen eine hohe Verarbeitungszeit da keine „normierten“ Werte vorliegen. Die Geschwindigkeit ist auch durch die Bildrate der Kamera(s) niedrig.	Abhängig von der Sensorik können sehr schnell sehr präzise Werte ermittelt werden. Es kann jedoch zu Einschränkungen bei der Art der Sensorwerte kommen (z.B. Position), speziell wenn nur die Sensorik im Gegenstand verwendet wird und auf externe Systeme (z.B. Sender/Empfänger mit bekannter, fixer Position) verzichtet wird.
Kalibrierung	Aufgrund der unterschiedlichen Anatomie muss die Kalibrierung für jeden Nutzer extra vorgenommen werden, teilweise muss sogar vor jeder Benutzung kalibriert werden.	Eine Kalibrierung kann meist für eine Benutzergruppe erfolgen.	Keine Kalibrierung erforderlich wenn nur die Bewegungen des Artefaktes und nicht z.B. die Gesten der Hand in denen der Gegenstand gehalten wird, von Interesse sind.
Anwendungsbereiche	Anwendung potenziell überall, da allgemeingültige Handparameter berechnet werden, die eine beliebige Weiterverarbeitung zulassen. Die Beschädigungsgefahr bei unsachgemäßer Handhabung ist hoch.	Anwendungsgebiete sind sehr oft sehr speziell und eingeschränkt. Die rein visuelle Erkennung ermöglicht auch den Einsatz an Orten mit besonderen Voraussetzungen, etwa in sterilen Räumen oder bei Vandalismusgefahr.	Verwendung potenziell überall da kein zusätzlicher Aufwand getrieben werden muss und die Gesten von vielen Systemen für unterschiedlichste Anwendungen im Umfeld genutzt werden können.

Tabelle 1: Gegenüberstellung der Vor- und Nachteile von Datenhandschuh, Methoden der Computer Vision und Sensorik in Artefakten [27][37]

Beim DataGlove erfolgt die Bestimmung durch die Messung des Magnetfeldes. Der Z-Glove hingegen verwendet eine preiswertere Ultraschallmessung. Obwohl die Technik sehr empfindlich und fehleranfällig ist, hat sich das Produkt, mangels Konkurrenz, in der Forschung weit verbreitet.

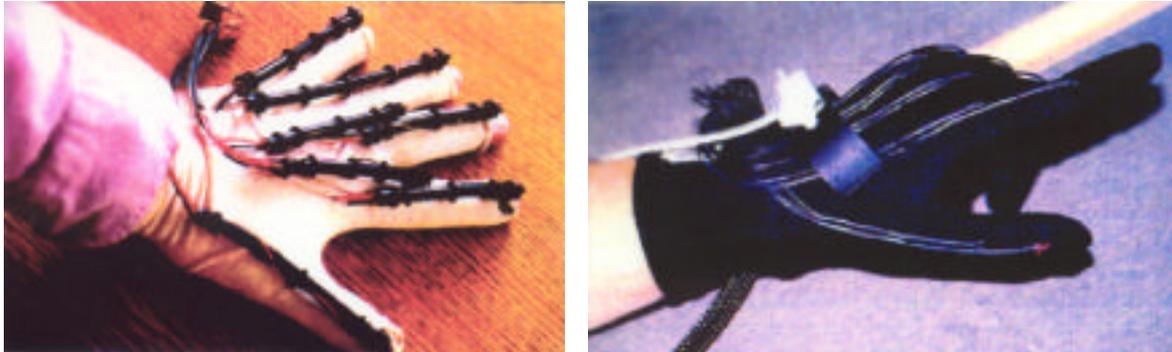


Abbildung 15: Sayre Glove, DataGlove (v.l.n.r.)

Es wurden noch weitere, teilweise sehr kostengünstige, Datenhandschuhe entwickelt. 1989 wurde von Mattel/Nintendo der PowerGlove produziert. Weitere bekannte Varianten sind der Space Glove (1991) und der 5th Glove (1995). Eine ausführliche Einführung in die Entstehung dieses Gebietes findet man von Sturman und Zeltzer in [48], eine aktuellere Betrachtung bietet LaViola in [27].

3.5.2 Anwendungen mit Datenhandschuhen

Derzeit werden Datenhandschuhe hauptsächlich in der virtuellen Realität zur Manipulation und Navigation angewendet. Es gibt jedoch auch einige Anwendungen die auf der Gestenerkennung durch solche Handschuhe basieren.

Fels und Hinton nutzen in ihrem Glove-TalkII Projekt [11] mehrere Sensoren, dabei spielen Datenhandschuhe (Cyberglove, ContactGlove) eine wesentliche Rolle. Es werden neuronale Netze verwendet um Zeichensprache, welche primär mit Handschuhen eingegeben wird, mit Hilfe eines Sprachgenerators in Sprache umzuwandeln. Aufgrund der Komplexität des Systems dauert die Trainingsphase relativ lange. Nach 100 Stunden konnte eine ausgewählte Testperson jedoch klar und verständlich sprechen, wenn auch sehr langsam (1,5 bis 3 Mal langsamer als normal). Insgesamt konnte ein besseres Ergebnis als bei konventionellen Text-zu-Sprache Systemen erzielt werden.

In [17][39] wird ein eigens entwickelter Datenhandschuh vorgestellt welcher auf Beschleunigungssensoren basiert (Abbildung 16). Die Sensoren sind an den Fingerspitzen befestigt. Aufgrund der Datenübertragung via Funk und der Spannungsversorgung mittels Batterien ist es prinzipiell möglich den Handschuh ohne Kabel zu betreiben, dadurch wird der Bedienkomfort wesentlich erhöht.

Es wurde ein Texteditor mit einem Befehlsvorrat von 28 statischen Gesten definiert, 26 Buchstaben und jeweils eine Geste für ein Leerzeichen und das Löschen. Nach wenigen Stunden konnte eine Testperson das gesamte Alphabet mit einer durchschnittlichen Geschwindigkeit von 0,91 Gesten pro Sekunde eingeben. 162 Gesten konnten in einem Versuch nacheinander erkannt werden. Durch die zusätzliche Betrachtung von dynamischen Gesten könnte man die Anzahl der unterscheidbaren Gesten weiter erhöhen und auch die Erkennungsgeschwindigkeit würde steigen, jedoch werden die Erkennungsalgorithmen hierzu komplexer.

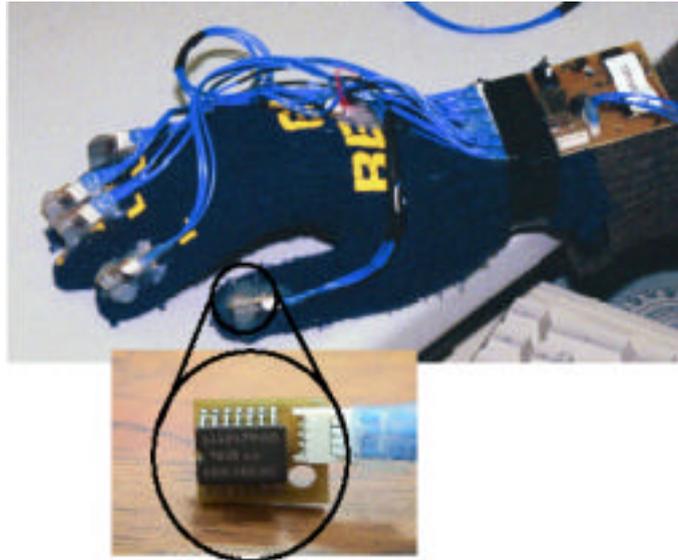


Abbildung 16: Drahtloser Datenhandschuh mit Accelerometern

3.6 Erkennung mittels Methoden der Computer Vision

Bei der Gestenerkennung mit den Methoden der Computer Vision, also der Analyse von Kamerabildern, ist es nicht zwangsweise erforderlich, dass der Benutzer etwas in der Hand hält oder anzieht. Diesen Vorteil erkaufte man sich jedoch mit einem großen algorithmischen Aufwand bei der Erkennung der Gesten. Der Benutzer kann sich frei bewegen, jedoch ist er noch immer durch den Bereich der Kameras eingeschränkt. Da die Kamera(s) meist stationär sind ist es nicht einmal möglich sich umzudrehen bzw. den Raum zu verlassen. Aufgrund des großen Rechenaufwandes sind mobile Systeme schwer zu realisieren. In [15] und [47] werden jedoch mögliche Ansätze hierfür gezeigt.

Die Komplexität der Gestenerkennung zeigt Nölker [37] anhand eines für den Menschen, vorausgesetzt er versteht die Gebärdensprache, einfachen Beispiels. Bei dem Versuch ein Video mit einer Person, die etwas in Gebärdensprache ausdrückt, durch eine Maschine analysierbar zu machen ergeben sich für den Rechner einige Probleme:

- Zunächst einmal muss die (sich bewegende) Hand von dem komplexen Hintergrund segmentiert werden.
- Das Tracking der Handposition muss oft relativ zur Umgebung geschehen.
- Die Bewegungstrajektorie der Hand muss analysiert werden.
- Die Handpostur und -orientierung muss auch bei schnellen Bewegungen mit zeitweiligen Verdeckungen erkannt werden.
- Zum Schluss müssen Bewegungstrajektorie, Handstellung und -orientierung integriert und ihre Gesamtbedeutung bestimmt werden.

Zu Beginn der Entwicklung wurden die Hände bzw. die Finger durch Handschuhe oder auch durch Farbpunkte, meist an den Fingerspitzen, markiert um eine Verfolgung der Hände zu erleichtern. Diese Markierungen sind nun nicht mehr notwendig, jedoch funktionieren die meisten Ansätze nur in sehr restriktiven Szenarien. Diese Einschränkungen beziehen sich primär auf die Art des Hintergrundes, die Erkennung eines kleinen Gestenvokabulars oder es können nur statische und keine dynamischen Gesten erkannt werden.

3.6.1 Einteilung basierend auf der Vorgehensweise

Nölker [37] unterteilt die Computer Vision basierten Methoden zur Bewegungserkennung in zwei Klassen: *merkmalbasiert* oder *modellbasiert*.

Merkmalbasierte Verfahren

Bei den merkmalsbasierten Verfahren wird eine Aussage über die Handgeste direkt und alleine aus dem Bild der Hand gewonnen. Nach einer Vorverarbeitung wird das Bild zunächst in Merkmale transformiert, aus denen im nächsten Schritt die Handpose berechnet wird.

Mögliche 2D-Bildmerkmale beinhalten Eigenschaften auf relativ niedriger Bildebene, wie beispielsweise die Kontur oder die Silhouette. Merkmale können jedoch auch Parameter sein, die das Bild auf einer höheren Bedeutungsebene beschreiben, also etwa die Position von speziellen Landmarken der Hand oder auch Bewegungsparameter.

Bei den Verfahren diese Merkmale weiterzuverarbeiten gibt es wiederum eine große Anzahl von Möglichkeiten, hierzu gehören etwa Hidden-Markov-Modelle, Entscheidungsbäume oder neuronale Netze.

Bei der überwiegenden Mehrzahl der Verfahren findet eine anwendungsspezifische Klassifikation in eine feste Anzahl von „bekannten“ Gesten statt, die in der Anwendung zum Beispiel als einfache Kommandosprache eingesetzt werden. Verfahren, die allgemeine Handparameter als Ergebnis liefern, sind hingegen selten [37]

Modellbasierte Verfahren

Bei modellbasierten Verfahren wird aus Bildern einer Hand von einer oder mehreren Kameras durch Abgleich mit einem 3D-Handmodell die beste Übereinstimmung gesucht. Dies geschieht entweder durch Vergleich mit dem gesamten 3D-Modell oder durch Tracking beginnend mit einer bekannten Startposition und/oder -geste der Hand. Dieses Verfahren ist natürlich nicht auf die Hand beschränkt, jedoch wird aufgrund der Möglichkeiten der Hand meist diese verwendet.

Modellbasierte Verfahren arbeiten in der Regel auf Bildern ohne visuelle Markierungen, da meistens der Abgleich mit der gesamten Hand stattfindet. Voraussetzung für eine gute Übereinstimmung zwischen Vorgabe und erkannter Gesten ist eine sorgfältige Modellierung der menschlichen Hand mit allen Freiheitsgraden und den Bewegungsmöglichkeiten sowie den Einschränkungen. Die variablen Parameter des Modells sind gewöhnlich die Gelenkwinkel, Orientierung und Position der Handfläche, seltener die statischen Größen der Hand, wie Fingerbreite, Segmentlänge usw.. [37]

Die Modelle lassen sich in zwei Klassen unterteilen: volumetrische (oder Oberflächen-) Modelle und Skelettmodelle.

Volumetrische Modelle findet man vorrangig im Bereich der Computeranimation - und durch die Entwicklung der Rechner nun auch in Anwendungen der Computer Vision. Der Fokus liegt hier auf einer möglichst detaillierten Beschreibung der äußeren Erscheinung der Hand, daher werden die komplexen 3D-Oberflächen etwa durch NURBS oder Splines modelliert. Dadurch wirken die Modelle zwar sehr realistisch, sind jedoch zu komplex, um in Realzeit wiederzugeben zu sein.

Als *Skelettmodell* bezeichnet man solche Modelle, deren Segmente als Linien mit einer gegebenen Länge dargestellt sind. Die Anzahl der Parameter zur Einstellung der Gelenkwinkel ist oft reduziert. Einsatz finden die Skelettmodelle insbesondere bei der rein mathematischen Beschreibung zur Bewegungsanalyse der Fingergelenke, da der Rechenaufwand zur Generierung vergleichsweise gering ist.

Ein Kompromiss zwischen realistischer Darstellung und geringer Rechenzeit sind 3D-Modelle, die aus einfachen, dreidimensionalen geometrischen Strukturen (Zylinder Ellipsoid, ...) aufgebaut sind. Durch das Zusammenfügen von mehreren einfachen Objekten lassen sich komplexe Strukturen, wie etwa die Hand, gut approximieren. Hier ist die Anzahl der notwendigen Parameter zur Beschreibung relativ gering, sodass die Berechnung des Modells in Echtzeit stattfinden kann. [37]

3.6.2 Zeitliche Entwicklung und Anwendungen

Davis und Shah [10] entwickelten bereits 1994 ein System mit dem insgesamt sieben unterschiedliche dynamische Gesten mithilfe eines modellbasierten Verfahrens erkannt werden konnten. Die Fingerspitzen wurden farblich gekennzeichnet um eine Verfolgung zu ermöglichen. Somit ist die Position der Fingerspitzen immer bekannt und es kann aus den Bewegungen eine Trajektorie bestimmt werden, welche die dynamische Geste beschreibt. Um dies einfacher zu gestalten muss jedoch immer von einer fix definierten Startposition ausgegangen werden. In diese Position muss folglich nach jeder Geste wieder zurückgekehrt werden, da sonst keine weiteren Gesten mehr erkannt werden können.

Um die Erkennung der amerikanischen Zeichensprache geht es 1995 in einem System von Starner [46]. Ausgehend von den Informationen einer Kamera sollen Sätze der amerikanischen Zeichensprache erkannt werden. Die Zeichen der Sätze bestehen teilweise auch aus Gesten welche mit beiden Händen durchgeführt werden, somit müssen bei der Erkennung ständig beide Hände verfolgt werden. Die Position, Orientierung und Exzentrizität der umhüllenden Ellipsen der beiden Hände dienen als Eingabeparameter für das Hidden-Markov-Modell. Da keine genauere Verfolgung, etwa die Fingerstellung, durchgeführt wird ist das System echtzeitfähig. Mit einem Zeichenvorrat von insgesamt 40 Gesten wurden zufällige Testsätze nach einer einfachen Grammatik gebildet. Mit der Verwendung von zwei unterschiedlich farbigen Handschuhen, zur einfacheren und fehlerfreien Bestimmung der linken und der rechten Hand, und dem Wissen über die erlaubte Grammatik konnte das System eine Erfolgsquote von 97% erreichen, ohne der Verwendung der Grammatik 91%. Probleme bereiten noch bedeutungslose Gesten, da diese nicht herausgefiltert werden können, weiters durfte sich die Testperson möglichst wenig bewegen um entsprechend gute Erkennungsraten zu erzielen.

Um das Problem, an eine bestimmte Stelle vor einer Kamera und auf den damit verbundenen stationären Einsatz beschränkt zu sein, zu lösen, wurde eine mobile Version entwickelt. [47] Hierzu wurde am Rand einer Kappe eine Kamera montiert, welche die Hände des Benutzers aufzeichnet (Abbildung 17). Hierbei wurde auch versucht die Hand aufgrund ihrer Hautfarbe und nicht aufgrund der Farbe der Handschuhe zu erkennen. Trotz der daraus resultierenden zusätzlichen Probleme verschlechterte sich die Erkennungsrate im Testfall mit der Grammatik kaum.

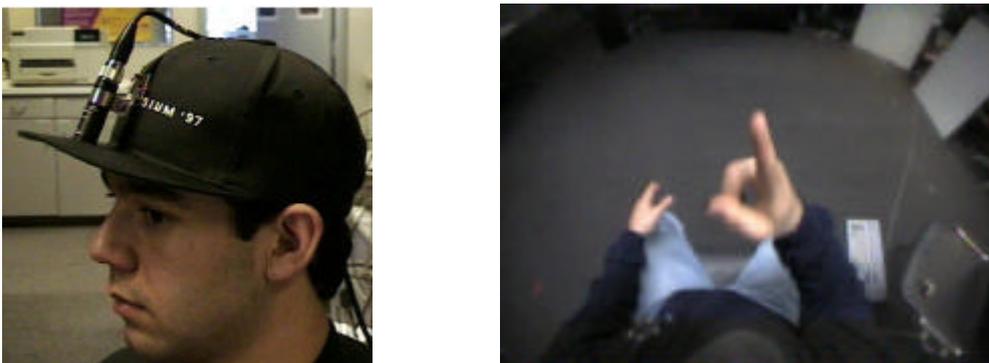


Abbildung 17: Kappe mit montierter Kamera bzw. Bild von dieser Kamera

Ebenfalls eine tragbare Lösung für ein Handverfolgungssystem zeigen Hamette et al. [15]. In diesem, sehr hardwarenahen Ansatz, wird versucht den eingeschränkten Möglichkeiten von tragbaren Systemen Rechnung zu tragen. Nicht zuletzt aufgrund der eingeschränkten Ressourcen muss die Hand in einer fixen Entfernung zur Kamera sein, auch die Startposition ist vorgegeben. Die verhältnismäßig kleine Hardware arbeitet vollständig autonom. Als Hauptanwendung wird eine Mausfunktionalität gezeigt, die mit dem Finger gesteuert wird.

Einen Überblick über die Merkmale und Eigenschaften, inklusive der damit realisierten Anwendungen, von rund 40 Mehrkamarasystemen zur Gestenerkennung zeigen Kohler und Schröter. [26]

4 Funktechnologien

Mittlerweile gibt es eine Vielzahl an Technologien zur drahtlosen Datenübermittlung. Fast alle der Ansätze sind auf ein bestimmtes Anwendungsgebiet ausgelegt, wobei sich die Bereiche an den Grenzen, und somit die damit verbundenen Anwendungen, teilweise überschneiden.

Es wurden nur lokale Kommunikationssysteme in den Überblick aufgenommen, somit fehlen nationale und globale Technologien wie GSM, GPRS und UMTS.

Im Folgenden werden die wesentlichen Eigenschaften aktuell verfügbarer Funktechnologien vorgestellt. Tabelle 2 basiert auf [14] und zeigt zentrale Eigenschaften der Technologien in einem kurzen Überblick.

	WLAN	Bluetooth	ZigBee	Proprietärer Funk	Infrarot
Frequenzbereich	802.11 b/g 2,4 GHz 802.11 a 5 GHz	2,4 GHz	868 MHz (Europa) 902 - 928 MHz (Amerika) 2,4 GHz (weltweit)	Freie Frequenzbänder z.B. 902 – 928 MHz	Infrarot 800-900 nm
Datenrate	11 Mbit/s 54 Mbit/s	1 Mbit/s	20 kbit/s 40 kbit/s 250 kbit/s	10 - 200 kbit/s	20 - 40 kbit/s 115 kbit/s 4 & 16 Mbit/s
Reichweite	100 Meter 50 Meter	10 – 100 Meter	10 – 100 Meter	10 Meter – 15 Kilometer	1 – 9 Meter Sichtkontakt
Topologie	Ad-hoc, Point to multipoint	Ad-hoc, piconets	Ad-hoc, star, peer to peer, mesh	Point to point	Point to point
Komplexität	Hoch	Hoch	Niedrig	Sehr niedrig	Niedrig
Stromverbrauch	Hoch	Mittel	Sehr niedrig	Niedrig	Niedrig
Anwendungen	WLAN Zugang	Drahtlose Headsets, PC-PDA-Laptop Verbindungen	Haustechnik, Sensor-netzwerke,	Kodierte Fernbedienungen, allgemeine Datenübertragung	Fernbedienungen, PC-PDA-Laptop Verbindungen

Tabelle 2: Überblick über aktuelle lokale Funktechnologien

4.1 WLAN (802.11)

Wireless-LAN (WLAN) basiert auf dem 1997 vom IEEE definierten Standard 802.11, die Wesentlichen mit diesen Standard zusammenhängenden Erweiterungen sind in Tabelle 3 angeführt.

IEEE-Standard	Geschwindigkeit	Frequenz	Anmerkung
802.11	1 bzw. 2 Mbit/s	2,4 GHz	1997 standardisiert
802.11a	Bis zu 54 Mbit/s	5 GHz	1999 standardisiert
802.11b	5,5 bzw. 11 Mbit/s	2,4 GHz	Derzeit dominierender Standard
802.11e			QoS (Quality of Service)
802.11f			Inter Access Point Protocol: Roaming
802.11g	Bis zu 54 Mbit/s	2,4 GHz	
802.11h	Bis zu 54 Mbit/s	5 GHz	Europäische Lösung für 802.11a
802.11i			Sicherheit: Authentifizierung

Tabelle 3: Wichtige WLAN Standards [29]

4.1.1 Topologie

WLANs können in zwei verschiedenen Topologien betrieben werden. Im Ad-hoc Modus, oder auch Independent Basic Service Set (IBSS, Abbildung 18), kommunizieren zwei oder mehrere Endgeräte direkt miteinander.

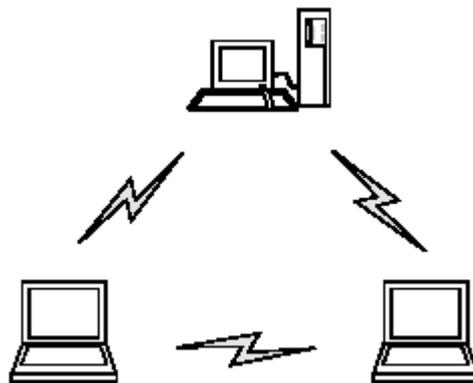


Abbildung 18: Independent Basic Service Set [5]

In den meisten Fällen wird ein WLAN im Infrastruktur-Modus betrieben, d. h. die Kommunikation der Clients erfolgt über eine zentrale Funkbrücke, den so genannten Access-Point. Hierbei wird von einem Basic-Service-Set (BSS) gesprochen.

Über den Access-Point erfolgt auch die Verbindung in kabelgebundene LAN-Segmente (Abbildung 19). Es können somit mehrere BSS gekoppelt werden. Das Gesamtsystem wird als „Extended Basic Service Set“ (ESS) bezeichnet, das koppelnde Netzwerk zwischen den BSS wird Distribution System (DS) genannt. [5][29]

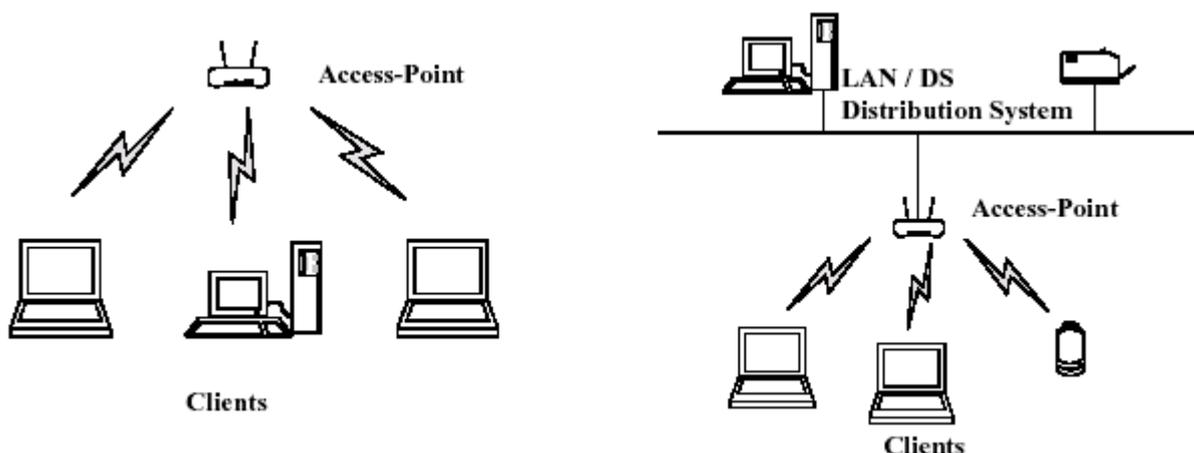


Abbildung 19: Infrastruktur Modus: Basic Service Set (links), Extended Basic Service Set (rechts) [5]

Der Infrastruktur-Modus lässt mehrere Einsatzvarianten zu:

- Mittels mehrerer Access-Points können überlappende Funkzellen installiert werden, sodass beim Übergang eines Clients in die nächste Funkzelle die Funkverbindung aufrecht erhalten werden kann ("Roaming"). Auf diese Weise können große Bereiche flächendeckend versorgt werden. Die Reichweite einer Funkzelle ist extrem abhängig von den Umgebungsbedingungen und liegt im Bereich von ca. 10 - 150 Meter.
- Zwei Access-Points können auch als Brücke (Bridge) zwischen zwei leitungsgebunden LANs eingesetzt werden. Ebenso ist der Einsatz eines Access-Points als Relaisstation (Repeater) zur Erhöhung der Reichweite möglich.
- Bei der Verwendung von Richtantennen an den Access-Points kann ein WLAN auch zur Vernetzung von Liegenschaften eingesetzt werden. Hier können Reichweiten im Kilometerbereich erreicht werden. [5][29]

4.1.2 Funkschnittstelle

Fast alle der in Europa zugelassenen WLAN-Systeme nach 802.11 und 802.11b nutzen das ISM-Frequenzband (Industrial-Scientific-Medical, siehe auch Abbildung 24: Übersicht der ISM Frequenzbereiche) zwischen 2,4 und 2,48 GHz, das gebührenfrei und ohne zusätzliche Genehmigung verwendet werden kann. Die Sendeleistung ist auf maximal 100 mW EIRP (Effective Isotropic Radiated Power) begrenzt.

Systeme des Standards 802.11 übertragen die Daten mit einer Rate von 1 bzw. 2 Mbit/s. Es wird ein Bandspreizverfahren eingesetzt und das Signal wird entweder mittels Frequency Hopping Spread Spectrum (FHSS) (1 Mbit/s) oder Direct Sequenz Spread Spectrum (DSSS) moduliert. Der Standard 802.11a bedient sich des Orthogonal Frequency Division Multiplexing (OFDM). Die aktuell gebräuchlichen Systeme nach Standard 802.11b verwenden hingegen das Verfahren High Rate Direct Sequenz Spread Spectrum (HR/DSSS). Die zu übertragenen Daten werden hierbei mit einem festen Code gespreizt d.h. über ein breites Frequenzband ausgedehnt, um die Übertragung unempfindlicher gegenüber Störungen zu machen.

Der Zugriff auf den Funkkanal erfolgt, wie bei allen Systemen der 802.11 Standards, nach einem zufallsgesteuerten Verfahren, genannt Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA). Die Brutto-Datenübertragungsrate beträgt bei 802.11b maximal 11 Mbit/s. Die Übertragungsraten können, wie bei allen Systemen der 802.11 Standards, nicht garantiert werden, sie hängen ab von der Anzahl der Clients und der Qualität der Funkübertragungsstrecke.

Im 2,4 GHz Frequenzbereich stehen in großen Teilen Europas 13 Frequenzkanäle mit einem Frequenzabstand von 5 MHz für die Funkübertragung nach 802.11b zur Verfügung. Bei einer Kanalbandbreite von ca. 22 MHz können jedoch nur maximal 3 Kanäle gleichzeitig überlappungsfrei genutzt werden.

802.11a und zukünftige 802.11h Systeme nutzen den 5 GHz Bereich. Im Frequenzbereich von 5,15 bis 5,35 GHz und bei 5,47 bis 5,725 GHz sind in Europa insgesamt 19 Kanäle in einem Abstand von 20 MHz freigegeben worden. Bei einer Kanalbandbreite von 20 MHz werden hier direkt benachbarte Kanäle nicht gestört. [5][29]

4.2 Bluetooth

Bluetooth wurde 1998 von der Bluetooth Special Interest Group (SIG) ins Leben gerufen. Die Designziele waren: Ersatz von Kabeln, preiswerte Bluetooth Module, niedriger Stromverbrauch und geringe Größe.

4.2.1 Topologie

In der Bluetooth Spezifikation wurden 2 unterschiedliche Topologien definiert: *Piconet* und *Scatternet*.

Ein Piconet besteht aus einem Master und bis zu sieben aktiven Slaves, insgesamt können jedoch 255 Geräte zu einem Piconet zusammengefasst werden. Eine mögliche Struktur ist in Abbildung 20 ersichtlich.

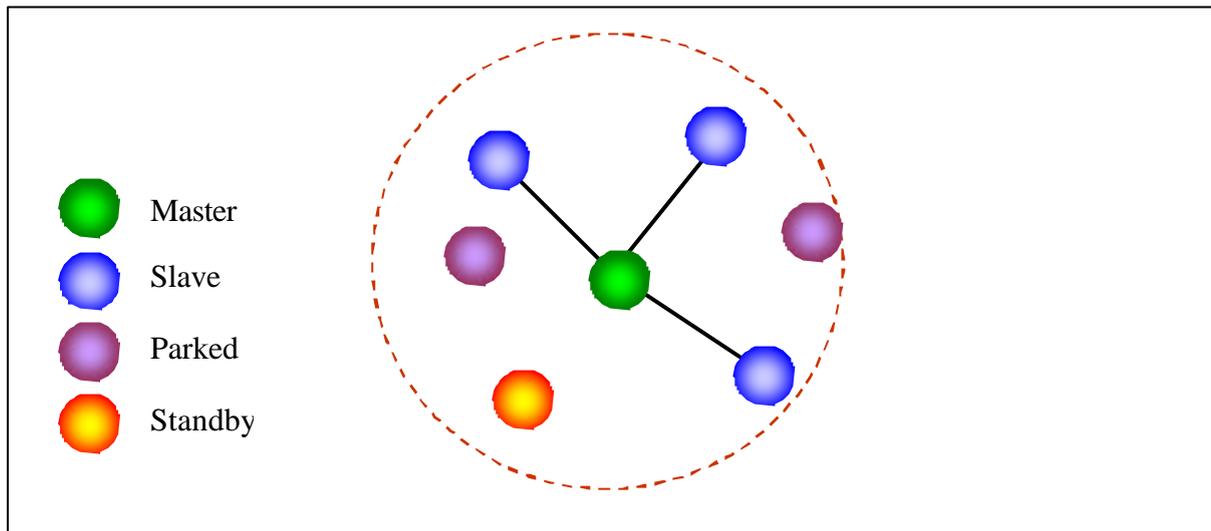


Abbildung 20: Bluetooth Piconet

Die Kommunikation erfolgt immer über den Master, eine direkte Verbindung zwischen zwei Slaves ist nicht möglich. Somit ergibt sich eine point to point Kommunikation zwischen Master und Slave, oder auch eine Multicast Verbindung vom Master zu allen Slaves. Die gesamte Datenrate in einem Piconet beträgt 1 Mbit/s.

Die Unterscheidung zwischen Master und Slave ist für die Anwendungen fast immer bedeutungslos und bezieht sich nur auf die Rolle im Netzwerk. [30]

In der Bluetooth Spezifikation [3] werden 3 unterschiedliche Hauptzustände definiert, in denen sich ein Gerät befinden kann:

- *Standby*: Der Standardzustand. Das Gerät ist kein Teilnehmer an einem Piconet.
- *Connection*: Es wurde eine Verbindung zwischen Master und Slave aufgebaut und eine bidirektionale Datenübertragung ist möglich.
- *Park*: Wenn ein Slave am Piconet nicht weiter teilnehmen muss, jedoch im Piconet verbleiben will kann er vom Master in den Park Zustand geschickt werden, hierdurch kann etwa der Strombedarf reduziert werden.

Wenn die Notwendigkeit besteht zwischen mehreren überlappenden Piconets zu kommunizieren, so kann das mithilfe eines Scatternet realisiert werden. Hierbei gehören ein oder mehrere Geräte mehreren Piconets an, siehe Abbildung 21. Die Kommunikation zwischen den Netzen erfolgt durch Weiterleitung durch die Geräte, welche sich in beiden Netzen befinden. Ein Gerät kann maximal in einem Piconet die Rolle des Masters übernehmen, in allen anderen ist er ein Slave. [30]

Insgesamt können bis zu 10 Piconets zu einem Scatternet zusammengeschlossen werden.

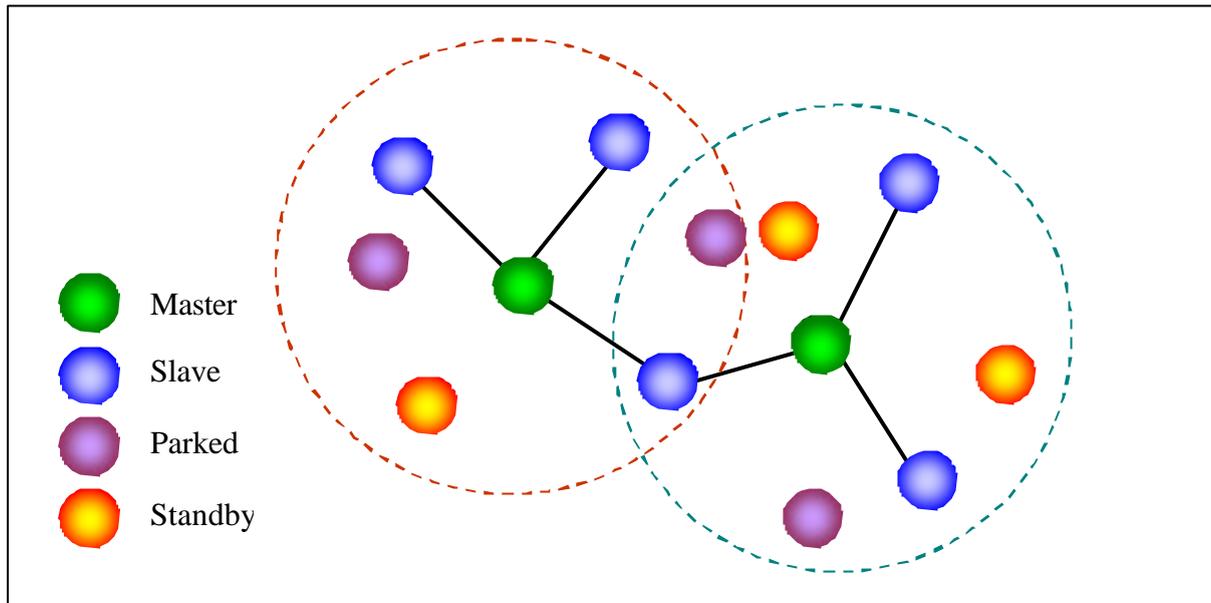


Abbildung 21: Bluetooth Scatternet

4.2.2 Technische Grundlagen

Bluetooth arbeitet im 2,4 GHz ISM (Industrial-Scientific-Medical, siehe auch Abbildung 24: Übersicht der ISM Frequenzbereiche) Frequenzband. Insgesamt stehen 79 Kanäle mit einer Bandbreite von je einem Megahertz zur Verfügung.

Als Modulationsverfahren kommt Gaussian Frequency Shift Keying (GFSK) zum Einsatz. Da hierbei die Frequenz verändert wird ist diese Modulationsart gegen Rauscheinflüsse relativ unempfindlich.

Die Übertragung der Datenpakete erfolgt zeitschlitzgesteuert in Verbindung mit einem Frequency Hopping Spread Spectrum (FHSS) Verfahren, wobei das Nutzsignal über sich permanent abwechselnde Trägerfrequenzen übertragen wird. Insgesamt sind fünf verschiedene Sprungfolgen spezifiziert, wobei die ersten vier unter anderem für die Suche nach Geräten verwendet werden und aus fest vorgegebene Sequenzen mit jeweils 32 Frequenzen bestehen. Für die eigentliche Datenübertragung wird aus der Geräteadresse des Masters eine Sprungfolge aus allen 79 Kanälen errechnet. Somit muss ein Gerät das sich im Parked Zustand befindet noch immer mit dem Master synchronisiert bleiben um wieder am Datenaustausch teilnehmen zu können. [30]

Im normalen Betrieb wechselt die Kanalfrequenz 1600 Mal pro Sekunde, damit ist jeder Zeitabschnitt im normalen Betrieb 625 μ s lang, nach Ablauf dieser Zeit wechselt die Frequenz wieder. Durch diese sehr geringe Dauer kann die Auswirkung von Störungen, die in einem schmalen Frequenzband auftreten, eingeschränkt werden. Wenn es auf einer bestimmten Frequenz zu Fehlern kommt, z.B. durch andere Funktechnologien wie WLAN oder auch durch Mikrowellenöfen, können die Daten kurze Zeit darauf auf einer anderen Frequenz gesendet werden. Dadurch wird erreicht, dass die Unterbrechungen der Verbindung sehr kurz gehalten werden kann. Die Frequenzfolge wiederholt sich im Datenübertragungsmodus alle 23,3 Stunden von neuem.

Diese Frequenzsprünge wirken zusätzlich wie eine Art Verschlüsselung, da sie das Abhören einer solchen Übertragung enorm erschweren. Vor allem aber erlaubt das FHSS Verfahren den gleichzeitigen Betrieb mehrerer Piconetze im selben räumlichen Überdeckungsbereich. Obwohl es zu einzelnen Kollisionen kommen wird, wenn kurzzeitig mehrere Piconetze auf derselben Frequenz kommunizieren, erhöht sich der durchschnittliche Gesamtdurchsatz [30]

Anhand der Sendeleistung lassen sich die Bluetoothgeräte in drei Klassen einteilen, diese sind in Tabelle 4 angeführt.

Leistungsklasse	Max. Ausgangsleistung	Reichweite
1	100 mW (20 dBm)	Bis zu 100m
2	2,5 mW (4 dBm)	Bis zu 20m
3	1 mW (0 dBm)	Bis zu 10m

Tabelle 4: Leistungsklassen von Bluetoothgeräten

Bluetooth bietet zwei Arten der physikalischen Verbindung um sich an unterschiedlichen Anforderungen anzupassen:

- Synchronous Connection Oriented (SCO) Link
- Asynchronous Connection Less (ACL) Link

Bei einem SCO Link besteht zwischen Master und Slave eine synchrone verbindungsorientierte point to point Verbindung mit einer maximalen Datenrate von 432,6 bit/s in beide Richtungen. Durch die vom Master exklusiv reservierten Zeitschlitze sind zeitkritische Übertragungen, wie etwa Sprachübertragungen, möglich. Ein wiederholtes Senden von SCO Paketen ist im Standard nicht vorgesehen. Ein Slave kann maximal drei SCO Links zu einem einzigen und zwei SCO Links zu unterschiedlichen Mastern verwalten.

Ein ACL Link bietet im Gegensatz zum SCO Link eine asynchrone verbindungslose Verbindung. Die maximale Datenrate beträgt hierbei 723,2 kbit/s in die eine und 57,6 kbit/s in die andere Richtung. Zwischen dem Master und allen Slaves, die ein Piconet bilden, wird in einer point to multipoint Verbindung kommuniziert. Hierfür werden die Zeitschlitze verwendet die noch nicht für SCO Links bestimmt sind. Der Master kann mittels ACL Links beliebige, am Piconetz beteiligte Slaves ansprechen, zwischen einem Master und einem Slave kann jedoch nur ein einziger ACL Link existieren. Verlorengegangene Pakete werden im Gegensatz zu SCO Links wiederholt gesendet. Ein einziger Master kann mehrere Verbindungen halten, die aus einer ACL und SCO Kombination bestehen. [3][30]

4.3 ZigBee

Der Standard IEEE 802.15.4 bietet eine Lösung für niedrige Datenraten mit sehr niedrigen Strombedarf und gleichzeitiger niedriger Komplexität. Aufgrund der Betriebszeit mit handelsüblichen Batterien von mehreren Monaten und sogar mehreren Jahren bieten sich Anwendungen wie die Haustechnik, drahtlose Sensornetzwerke, interaktive Spielsachen und Fernbedienungen an.

Der Zusammenhang zwischen IEEE 802.15.4 und ZigBee ist in Abbildung 22 dargestellt. IEEE 802.15.4 definiert die beiden hardwarenahen Schichten MAC (media access controller) und PHY (physical). Die darauf aufbauenden Schichten Netzwerk/Sicherheits-Schicht bzw. Das Anwendungsframework werden von der ZigBee Allianz [59] definiert. Die Anwendungsprofile werden von der ZigBee Allianz organisiert.

Der IEEE 802.15.4 Standard definiert zwei PHY Schichten. Eine der Schichten betrifft das 2,4 GHz Band, in dem 16 Kanäle verwendet werden. Hierbei kann durch O-QPSK (Orthogonal Quadratur Phase Shift Keying) eine Datenrate von 250 kbit/s erzielt werden. Die zweite Schicht betrifft zehn Kanäle bei 902 – 928 MHz (Amerika) bzw. einen Kanal bei 868 – 870 MHz (Europa). Durch BPSK (Biphase Shift Keying) wird eine Datenrate von 40 kbit/s

(Amerika) bzw. 20 kbit/s (Europa) erreicht. In allen drei Frequenzbereichen können Geräte ohne Lizenzkosten betrieben werden.

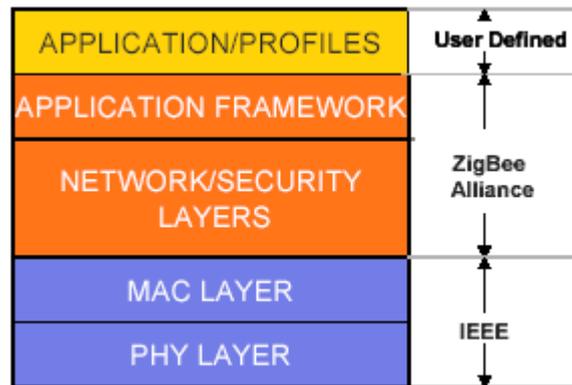


Abbildung 22: IEEE 802.15.4 Stack [8]

Die MAC Schicht ist für die Verbindung zum, und das Trennen vom Netzwerk zuständig. Als Zugriffsmethode wird Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) verwendet. Zusätzlich steht noch ein Mechanismus für eine Kommunikation mit hoher Priorität zur Verfügung.

Schicht drei (Network) stellt insgesamt drei Netzwerktopologien zur Verfügung: star, mesh und cluster tree. Star Netzwerke sind sehr einfach und gewährleisten selbst im Batteriebetrieb eine sehr lange Laufzeit. Mesh, oder peer to peer, Netzwerke ermöglichen gute Skalierbarkeit und hohe Ausfallsicherheit, da sie mehr als einen Pfad durch das Netzwerk zur Verfügung stellen. Ein cluster tree ist ein hybrider Ansatz zwischen star und mesh indem die Vorteile der beiden Topologien, Ausfallsicherheit und die Unterstützung von batteriebetriebenen Knoten, verknüpft werden. Eine, aus den Grundstrukturen zusammengestellte, mögliche Topologie ist in Abbildung 23 zu sehen.

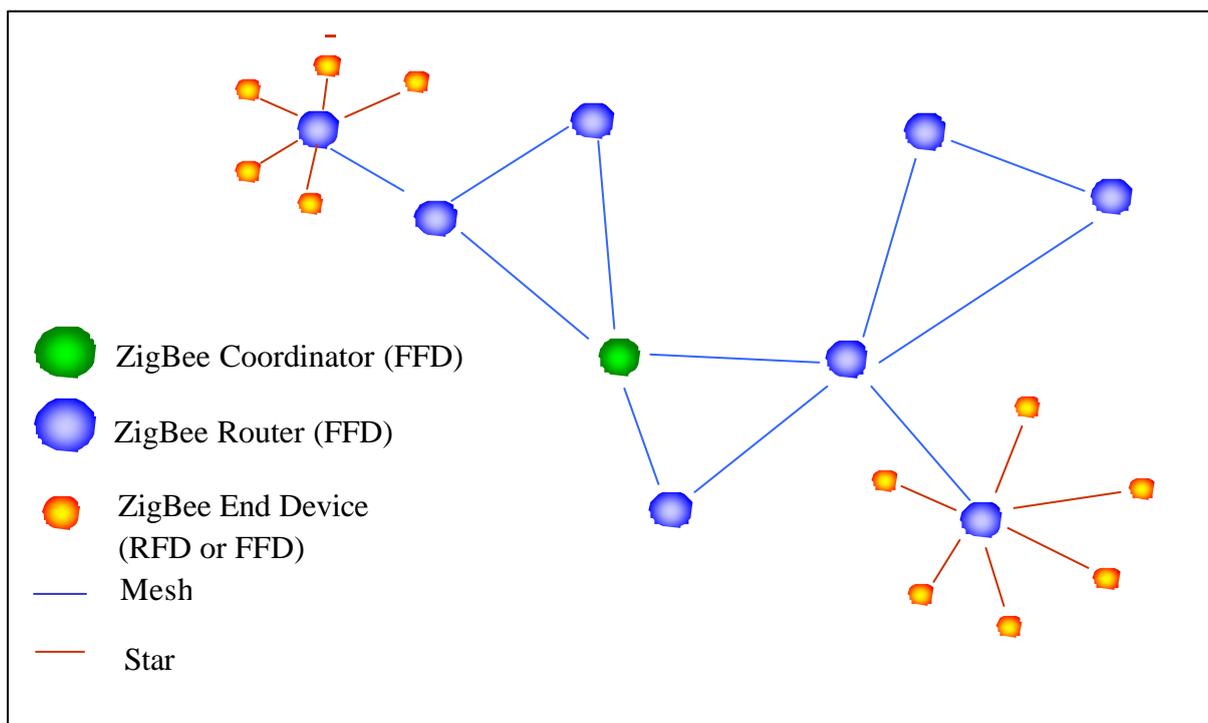


Abbildung 23: ZigBee Netzwerk Topologie [24]

In ZigBee gibt es eine physikalische Unterscheidung zwischen den Knoten um eine kostengünstige Herstellung zu ermöglichen. Es wird zwischen reduced function device (RFD) und full function device (FFD) unterschieden. In einem Netzwerk muss zumindest ein FFD vorhanden sein um die Rolle des Netzwerkkoordinators zu übernehmen. Die einzelnen Knoten haben folgende Eigenschaften:

- Full function device (FFD)
 - Kann in jeder Topologie verwendet werden.
 - Kann die Rolle des Netzwerkkoordinators und eines ZigBee Routers übernehmen.
 - Kommunikation mit jeden anderen Knoten (FFD/RFD) ist möglich.
 - Normalerweise netzbetrieben.

- Reduced function device (RFD)
 - Beschränkt auf die star Topologie.
 - Kann kein Netzwerkkoordinator werden.
 - Kommunikation nur mit Netzwerkkoordinator möglich.
 - Einfache Implementierung, geringe Anforderungen an RAM und ROM.
 - Normalerweise batteriebetrieben.

Ein RFD kann somit lediglich nach verfügbaren Netzen suchen und mit dem Netzwerkkoordinator in Abhängigkeit der Anwendung Daten austauschen. Durch eine Schlaf-Funktion kann der Strombedarf weiter reduziert werden.

In ZigBee gibt es neben der physikalischen Unterscheidung zwischen FFD und RFD auch noch eine logische Unterscheidung. Die logischen Rollen sind *Netzwerkkoordinator*, *Router* und *Endgerät*. Die Rolle ist ein Teil der Application/Profiles Schicht. Pro Netzwerk muss es einen Netzwerkkoordinator geben, der ein Netzwerk initialisiert, die Kommunikation leitet und die Daten über das Netzwerk speichert. Der ZigBee Router nimmt am Netzwerk teil, indem er Nachrichten zwischen den verbundenen Knoten leitet. Die Endgeräte müssen immer ein Blatt in der Topologie darstellen.

Insgesamt können 65535 Netzwerke bestehen und aufgrund der 64 Bit IEEE Adresse gibt es eine entsprechend große Anzahl von Knoten.

Die ersten ZigBee Produkte werden für das vierte Quartal 2004 erwartet, wobei sich zu diesem Zeitpunkt die Anzahl der unterstützten Profile (drahtlose Beleuchtung und Steuerung) noch in Grenzen halten wird. [59]

4.4 Proprietärer Funk

Neben den standardisierten Funktechnologien gibt es noch spezielle, meist auf bestimmte Anwendungen abgestimmte, Datenübertragungsmöglichkeiten. Durch die Einschränkung auf eine Anwendung können einzelne Parameter, wie etwa der Stromverbrauch, die Reichweite, usw., optimiert werden.

Um einen lizenzfreien Betrieb zu ermöglichen arbeiten diese Funkmodule in den unterschiedlichen ISM (Industrial-Scientific-Medical) Frequenzbereichen, siehe Abbildung 24. [45]

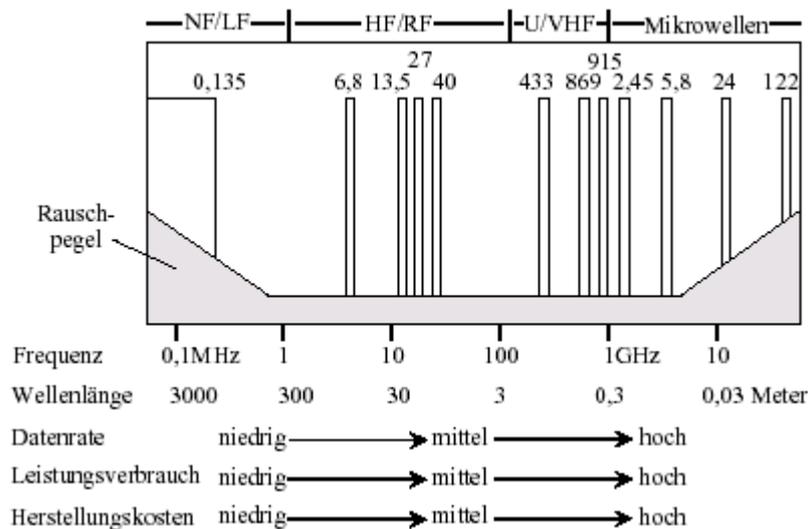


Abbildung 24: Übersicht der ISM Frequenzbereiche

Die am häufigsten verwendeten Bereiche liegen zwischen 433 MHz und 2,4 GHz, da hier ein guter Kompromiss zwischen Datenraten, Leistungsverbrauch und Herstellungskosten gefunden werden kann.

Neben den verwendeten Frequenzen sind auch weitere Parameter der Realisierung wie die Modulationsverfahren, Verhalten bei Kollisionen, Datenrate, Latenz, Störungsverhalten, Antennenbauform, Spannungsversorgung usw. sehr flexibel, somit ist eine allgemeine Betrachtung kaum möglich.

Als führender Hersteller auf dem Gebiet der Breitbandfunktechnologie kann Radiometrix [42] genannt werden. Radiometrix bietet beispielsweise das Modul BiM2-433 an, welches folgende wesentliche Eigenschaften hat:

- ISM Frequenzband bei 433 MHz
- Datenrate bis zu 160 kbit/s
- Reichweite bis 200 Meter im Freien bzw. 50 Meter in Gebäuden
- Stromverbrauch < 20 mA @ 3V
- Baugröße von 23 x 33 x 4 mm

4.5 Infrarot

Die Infrarotschnittstelle wurde ursprünglich als kabelloser Ersatz der seriellen Schnittstelle konzipiert, sie arbeitet bidirektional im Halbduplex-Verfahren mit Licht der Wellenlänge von 850 – 900 Nanometer. Heute wird hauptsächlich zwischen IrDA Data und IrDA Control unterschieden. Mittlerweile entsteht jedoch ein neuer Standard mit dem Namen AIR (advanced infrared). [5][19]

4.5.1 IrDA Data

Der IrDA Standard definiert ursprünglich eine Datenrate bis 115,2 kbit/s, die in der Version 1.1 unter Aufrechterhaltung der Kompatibilität auf bis zu 4 Mbit/s erhöht wird. Version 1.2 beinhaltet eine Low-Power-Version, die mit reduzierter Sendeleistung, Datenraten bis zu 115,2 kbit/s erreicht, während die Version 1.3 den Low-Power-Betrieb auf 1,152 Mbit/s und 4 Mbit/s erweitert. Die aktuelle Version 1.4 bietet durch ratenabhängige Kodierung der Datenbits Datenraten bis 16 Mbit/s.

Die Reichweite hängt von der abgestrahlten Leistung ab, sie beträgt in den Low-Power-Betriebsarten 20 Zentimeter, bei normaler Leistungsabgabe 1 bis 2 Meter. Die Datenübertragung wird über einen CRC (Cyclic Redundancy Check) Mechanismus abgesichert. Für Datenraten bis 1,152 Mbit/s wird CRC-16 verwendet, darüber hinaus CRC-32.

Um eine Verbindung aufzubauen, müssen zwei Geräte mit ihren Schnittstellen so aufeinander ausgerichtet werden, dass eine direkte Sichtverbindung zustande kommt. Hindernisse und größere Distanzen sind mit IrDA nicht zu überbrücken. Jedoch ergeben sich aus dieser Betriebsart auch Vorteile, da Störungen nur von sehr hellen Lichtquellen oder direkter Sonneneinstrahlung ausgehen [5][19]

4.5.2 IrDA Control

IrDA Control ist ein Protokoll zur Kommunikation von Peripheriegeräten wie Tastaturen, Mäusen, Joysticks oder Fernbedienungen mit zentralen Hosts, wie z.B. PCs und Fernsehgeräten. Für diese Infrarotübertragung ist eine andere physikalische Bitübertragungsschicht definiert, mit Datenraten bis 75 kbit/s und Reichweiten bis 5 Metern, so dass IrDA Data und IrDA Control zueinander nicht kompatibel sind. IrDA Control arbeitet mit einem Zeitschlitzverfahren, so dass ein Host mehrere Peripheriegeräte gleichzeitig ansteuern kann. Um den Bewegungsraum zwischen Host und Peripheriegerät nicht unnötig einzuschränken, ist in der Spezifikation ein Arbeitswinkel von $\pm 50^\circ$ vom Host angegeben. [5][19]

4.5.3 Protokollarchitektur

Die IrDA Protokollarchitektur definiert über der physikalischen Bitübertragungsschicht eine für den Verbindungsaufbau zuständige IrDA-Link-Access-Protocol-Schicht (IrLAP). Hier wird zwischen den zwei Betriebszuständen NDM (Normal Disconnected Mode) und NRM (Normal Response Mode) unterschieden. Im nicht verbundenen Modus (NDM) wird nach Signalen möglicher Kommunikationspartner gesucht und gleichzeitig ein Informationssignal mit 9600 Bit/s ausgesendet. Bevor nach einem Gerät in Reichweite gesucht werden kann muss zumindest 500ms gewartet werden ob ein Signal empfangen wird um Kollisionen zu vermeiden. Durch diese Zeit wird auch die Latenz beim suchen neuer Geräte bestimmt. Wird ein kommunikationsfähiges Gerät entdeckt, wird zwischen den zwei Geräten eine Verbindung hergestellt (NRM). Daraufhin werden die jeweiligen Leistungsdaten der Geräte ausgetauscht. IrLAP sichert die Verbindung mit Hilfe von Fehlerkorrektur und erneutem Senden von Daten im Bedarfsfall.

Auf IrLAP aufsetzend sorgt als nächst höhere Schicht das IrDA Link Management Protocol (IrLMP) für Mehrfachzugriff, Kanalbereitstellung für Dienste und das Informationsmanagement. Neben den hier genannten verbindlichen Protokollschichten existieren optionale Protokolle, die anwendungsspezifisch den Datenaustausch steuern, etwa für Mobiltelefonen und PDAs. [5][19]

5 Technische Realisierung

Bereits in der Problemstellung ist eine Aufteilung in zwei große Teile, den Hard- und Softwareteil, ersichtlich. Die zentralen Anforderungen an die Hardware sind die Datenerfassung und -übermittlung an eine möglichst universell gehaltene und damit mit bzw. von vielen Geräten verwendbare Empfangseinheit. Bei der Software spielt die einfache Bereitstellung der erkannten Gesten und der relevanten Sensordaten für den Anwendungsentwickler eine große Rolle, weiters soll die Abhängigkeit zwischen der Software und der verwendeten Hardware möglichst gering gehalten werden.

Die Anforderung, dass keine zusätzliche Infrastruktur benötigt werden soll, führte zu großen Einschränkungen bei der Auswahl der Sensorik, da hierdurch wichtige Sensortechnologien für diese Anwendung weggefallen sind, weiters ergeben sich hierdurch Einschränkungen bei der Wahl der Funktechnologie.

5.1 Auswahl der Hardwarekomponenten

Aufgrund der Problemstellung ergeben sich einige wesentliche Kriterien für die Hardware. Das System soll einen autonomen Betrieb ermöglichen, dies bedeutet, dass die Daten des Sensors mittels Funk an eine Auswerteeinrichtung gesendet werden müssen, da keine drahtgebundene Datenübertragung möglich ist. Eine direkte Auswertung der Sensorrohdaten durch zusätzliche Hardware direkt beim Sensor erfolgt nicht, da hierdurch zusätzliche Einschränkungen in Kauf genommen werden müssten und, speziell in dieser Versuchsumgebung, derzeit sowieso noch ein Gerät mit ausreichender Rechenleistung zur Verfügung steht. Da bereits erkannte Gesten nicht direkt Auswirkungen auf das Objekt in dem die Hardware integriert ist haben müssen, muss die Informationen über die erkannten Gesten an eine weitere Station gesendet werden, somit ist trotz der Auswertung der Sensordaten durch einen Mikrocontroller direkt beim Sensor ein Funkmodul zur Übertragung der erkannten Gesten erforderlich. Weiters würde natürlich der Entwicklungsaufwand für die Hardware steigen. Neben der Datenübertragung erfordert der autonome Betrieb auch noch eine mobile Spannungsversorgung.

Bei der Auswahl der Komponenten wurde dem Preis eine untergeordnete Rolle zugeteilt, da es primär um die Realisierungsmöglichkeiten geht und nicht um eine unmittelbare weite Verbreitung, außerdem ändert sich das Preisniveau bei der Produktion von entsprechend großen Stückzahlen und erfordert eine eigene Bewertung. Die Wahl viel teilweise auf teurere Komponenten als für die Anwendung unbedingt erforderlich gewesen wären um den Prototypen auch noch in weiteren Szenarien verwenden zu können. Diese Vorgehensweise erlaubt es auch die Hardwareanforderungen für solch ein System einfacher zu bestimmen und nicht im Vorhinein zu beschränken, weiters kann bei Bedarf eine weitere Version mit dem Fokus auf den Preis realisiert werden.

Die Größe und das Gewicht spielten bei der Auswahl eine zentrale Rolle, da die gesamte Hardware in handlichen Gegenständen des täglichen Lebens Platz finden soll, dabei sollen die Gegenstände nicht unnatürlich schwer oder in sonst einer Eigenschaft beeinträchtigt werden. Die Form sollte hierbei möglichst flexibel gehalten werden um sich den vorhandenen Gegebenheiten bezüglich des Platzes in den Artefakten besser anpassen zu können.

Bei der Auswahl wurde mit der Sensorik begonnen, da diese die Anforderungen für die weiteren Teile festlegt.

5.2 Sensorik

Die Sensorik ist die wesentliche Komponente der gesamten Hardware, da hierauf die Bewegungserkennung und somit die gesamte Anwendung basiert. Aufgrund der an das System gestellten Anforderungen können die beiden für Bewegungserkennung am häufigsten verwendeten Sensoren, Kamera und Datenhandschuh, nicht verwendet werden.

Bei den Systemen zur Bestimmung der relativen Position mittels linearer Beschleunigungssensoren treten Probleme auf, da es bei niedrigen Beschleunigungswerten immer wieder dazu kommt, dass die resultierenden Werte stark driften, sie sind dadurch für einen robusten Einsatz kaum geeignet.

Weiters bieten sich Gyroskope an, hierbei wird die Drehbeschleunigung um eine Raumachse gemessen, dies bringt jedoch den Nachteil mit sich, dass nur Informationen über die Orientierung jedoch nicht über die Position vorhanden sind. Trotz dieser Einschränkung wurde diese Sensorklasse für diese Anwendung als sehr brauchbar eingestuft und folglich viel die Wahl auch auf einen Sensor dieser Klasse. Hieraus ergaben sich zwei mögliche Ansätze der Vorgehensweise. Die erste Möglichkeit ist die Kombination von mehreren einfacheren Sensoren, also beispielsweise drei Sensoren die jeweils die Drehbewegung um eine Achse messen. Diese Sensoren verfügen beispielsweise über eine zur Drehung proportionale Spannung als Ausgabewert, das bedeutet jedoch dass eine zusätzliche Vorverarbeitung oder zumindest eine Digitalisierung der Werte mittels AD-Wandler erfolgen muss, weiters müssen in Abhängigkeit der Sensoren zusätzliche Kompensationsmechanismen entwickelt werden, um die Fehler der Messung etwa durch die Temperaturabhängigkeit gering zu halten. Als Vorteile können die sehr kleine Bauform sowie der sehr geringe Stromverbrauch gesehen werden. Die zweite Möglichkeit sind Sensoren die mehrere einzelne Sensoren zu einem zusammenfassen und die Messwerte bereits etwas aufbereiten, die Ausgabe der Werte erfolgt meist über eine genormte Schnittstelle, wie etwa RS232 oder auch USB. Die Vorteile sind in der optimalen Abstimmung der Sensoren aufeinander, die bereits erfolgte Kompensation unterschiedlicher Störinflüsse und der Datenübertragung über die genormte Schnittstelle zu sehen. Diese Vorteile werden meist mit einem höheren Stromverbrauch, einem höheren Preis und teilweise auch mit einer größeren Bauform bezahlt. Aufgrund der Vorteile der zweiten Möglichkeit und auch um den Hardware Entwicklungsaufwand für den gesamten Aufbau in Grenzen zu halten wurde ein fertiges System mit mehreren Sensoren gesucht welches die Anforderungen bestmöglich erfüllt.

5.2.1 Auswahl - InertiaCube²

Aufgrund der Problemstellung ergeben sich sehr hohe Anforderungen an den Sensor. Die wesentlichen Eigenschaften die der Sensor besitzen muss sind:

- Orientierungsmessung aller drei Achsen
- Möglichst kleine Bauform und geringes Gewicht
- Geringer Stromverbrauch
- Relative Genauigkeit zumindest 1°
- Maximale Drehrate mindestens 600°/s
- Aktualisierungsrate von zumindest 60 Hz

Wesentliche Parameter von sechs Sensoren die aufgrund der Anforderungen an den Sensor in die nähere Auswahl gekommen sind, sind in Tabelle 5 kurz gegenübergestellt. Die Wahl fiel auf den InertiaCube² von InterSense (Abbildung 25), da dieser in Summe die gestellten Anforderungen am Besten erfüllt, obwohl der Stromverbrauch mit 100 mA im Vergleich zu den anderen Sensoren verhältnismäßig groß ist und somit die Dauer des autonomen Betriebs stark beeinflusst.

	Ascension 3D-Bird	Crossbow Technology AHRS	InterSense InertiaCube ²	InterSense IS-900	MicroStrain 3DM	Xsens MT9
						
Sensorik						
Messbereich Yaw/Pitch/Roll	±180° / ±180° / ±90°	360° / ±180° / ±90°	360° / 360° / 360°	360° / 360° / 360°	±180° / ±180° / ±70°	360° / 360° / 360°
Genauigkeit Yaw/Pitch/Roll	jeweils 2,5° RMS stat. bzw. 4° RMS dyn.	±2° / ±2,5° / ±2,5°	jeweils 1° RMS	0,4° / 0,4° / 0,4° RMS	1,0° / 0,33° / 0,93°	jeweils 1° stat. bzw. 3° RMS dynamisch
Auflösung Yaw/Pitch/Roll	0,2° / 0,2° / 0,2°	0,1° / 0,1° / 0,1°	0,01° / 0,01° / 0,01° RMS	0,05° / 0,05° / 0,05° RMS	0,1° / 0,1° / 0,1°	0,05° / 0,05° / 0,05° RMS
Max. Drehrate	1000°/s	200°/s	1200°/s	-	-	900°/s
Beschleunigung	-	10 G	500 g	-	-	5000 m/s ²
Aktualisierungsrate	160 Hz	100 Hz	180 Hz	180 Hz	30 – 45 Hz	
Temperatur	-	-40° - 70°C	0° - 50°C	-	-25° - 70°C	5° - 45°C
Sonstiges	Verzögerung 15ms bei 360°/s	vollstabilisierte Daten nach 60s	gültige Daten nach 50ms	Infrastruktur wird benötigt	Datenrate 9600 Bits/s	vollstabilisierte Daten nach 10s
Elektrische Daten						
Versorgungsspannung	5V DC	10 – 35V DC	6V DC	100 – 240V AC	5,3V DC	5,5V DC
Leistungsaufnahme	100 mA	< 4 W	100 mA	100 W	50 mA	40 mA
Ausgabeform	RS232	RS232, RS422	RS232	RS232	RS232, RS485	RS232, RS485
Mechanische Daten						
Abmessungen [mm]	34,04 x 27,43 x 25,40	119 x 114 x 124	27,86 x 24,00 x 28,58	425 x 305 x 101	43,18 x 63,50 x 17,78	39,00 x 54,00 x 28,00
Gewicht	28,4 g	-	25 g	-	26,3 g	40 g
Sonstiges						
Software & Besonderheiten	dll mit Beispiel- programmen	FAA Anforderungen, FAR 23 class 1, 2 und 3 aircraft	dll mit Beispiel- programmen	dll mit Beispiel- programmen	-	Messung von Temp., Beschleunigung, magnetisches Feld

Tabelle 5: Überblick über verfügbare Sensoren. Einträge mit "-" wurden vom Hersteller nicht angegeben.

Die wichtigsten technischen Daten [21] des Sensors sind:

- Messbereich aller drei Achsen jeweils 360°
- Genauigkeit aller drei Achsen jeweils 1° RMS
- Auflösung aller drei Achsen jeweils $0,01^\circ$ RMS
- Aktualisierungsrate: 180 Hz
- Maximale Drehrate: $1200^\circ/\text{s}$
- Latenz: 8 ms
- Versorgungsspannung: 6 V
- Leistungsaufnahme: 100 mA
- Ausgangsformat: RS232 mit 115,2 kBits/sec
- Datenübertragung und Spannungsversorgung mittels Kabel
- Abmessungen: 27,86 mm x 24,00 mm x 28,58 mm
- Gewicht: 25 g



Abbildung 25: InterSense InertiaCube²

Der Sensor ist, wie man nicht zuletzt am rund viereinhalb Meter langen Kabel sehen kann, nicht für mobile Anwendungen gedacht. Das soll jedoch, bis auf den verhältnismäßig hohen Stromverbrauch von 100 mA, kein Problem bereiten, da dieses Kabel nahe am Sensor durch eine entsprechende Funklösung ersetzt werden kann. Die sonstigen technischen Daten wie etwa die Genauigkeit oder auch die Auflösung übererfüllen die Anforderungen für die Bewegungserkennung, da es für den Menschen beispielsweise kaum möglich ist Gesten mit Abweichungen von weniger als 1° zu wiederholen. Weitere Eigenschaften wie die Aktualisierungsrate und die maximale Drehzahl können hingegen bei bestimmten Bewegungen ausgereizt werden.

Zur Kommunikation mit dem Sensor und der Auswertung der Sensordaten werden vom Hersteller Bibliotheken zur Verfügung gestellt, diese erlauben es jedoch nicht auf die Rohdaten der einzelnen im InertiaCube² verwendeten Sensoren zuzugreifen, sondern es wird bereits die Orientierung im Bereich $0..360^\circ$ angegeben.

Der größte Nachteil dieses Sensors liegt, neben dem Stromverbrauch, im Preis, aktuell wird ein Preis für den InertiaCube² pro Stück in der Größenordnung von 1400 Euro angegeben. Dies bedeutet, dass diese Lösung in dieser Form (mit dieser Sensorik) sicher keine weite Verbreitung findet. Für die Erhebung der Anforderungen, der Möglichkeiten aber auch der Grenzen von einem solchen System zur Bewegungserkennung spielt der Preis jedoch eine

untergeordnete Rolle, dies rechtfertigt auch den Einsatz der relativ teuren Sensorik für den Prototypen.

In einem ersten Versuch den Sensor autonom zu gestalten wurde das Kabel auf wenige Zentimeter abgeschnitten und mit einem neunpoligen SUB D Stecker (Standardstecker für RS232) versehen, dies sollte weiterhin den Betrieb des Sensors ohne zusätzliche Adapter ermöglichen. Mehrere Tests ergaben jedoch, dass die Sensorwerte durch diesen metallischen Stecker in unmittelbarer Nähe zum Sensor beeinflusst und somit verfälscht werden. Aus diesem Grund wurde ein kleiner Stecker mit geringem Metallanteil gesucht. In Abbildung 26 ist die aktuelle Version des Sensors mit gekürzten Kabel und einer zweireihigen Steckerleiste mit zehn Kontakten und zwei Millimeter Rastermaß (Teil der TCMD Modellreihe von Samtec) für die Datenübertragung und die Spannungsversorgung zu sehen.



Abbildung 26: InertiaCube² mit gekürzten Kabel und neuem Stecker

5.3 Kommunikation

Die Problemstellung erfordert eine einfache möglichst universelle Übermittlung der Sensordaten an eine Auswertestation. Insgesamt ergeben sich folgende Anforderungen an das Kommunikationsmodul:

- Möglichst kleine Bauform und geringes Gewicht
- Geringer Stromverbrauch
- Kompatibel mit möglichst vielen Auswertestationen
- Keine zusätzliche Infrastruktur (Sender usw.)
- Datenrate von mindestens 115,2 kBits/sec (aufgrund des gewählten Sensors)
- Idealerweise RS232 als Eingang (aufgrund des gewählten Sensors)
- Reichweite von zumindest 5-10 Meter

Basierend auf der sehr detaillierten Betrachtung der Funktechnologien in Abschnitt 4 lässt sich einfach beurteilen welche Technologie die gestellten Anforderungen am Besten erfüllt. Gegen WLAN spricht der zusätzliche Aufwand der notwendig ist um die Sensordaten zu übertragen, auch die Leistungsaufnahme ist hoch und die Datenrate kann bei weitem nicht ausgenutzt werden. IrDa ist alleine aufgrund der notwendigen Sichtverbindung zum Empfänger für die Anwendung nicht geeignet. Eine proprietäre Lösung würde auf Seiten des Sensors die Anforderungen sehr gut erfüllen, jedoch bedarf es hierbei eines speziellen Empfängers, somit eignet sich auch diese Technologie nicht optimal. Bei der Betrachtung von ZigBee und Bluetooth liegt von den technischen Eigenschaften für diese Anwendung gesehen

der Vorteil auf Seiten von ZigBee, aufgrund der mangelnden Verfügbarkeit der Geräte viel die Wahl jedoch auf Bluetooth.

5.3.1 Auswahl - connectBlue OEM Serial Port Adapter 33i

Aufgrund des Sensors der die Daten mittels RS232 zur Verfügung stellt wurde nach einem RS232-Bluetooth Konverter gesucht, es sollte also kein zusätzlicher Aufwand für die Hardwareentwicklung erforderlich sein.

Mehrere Anbieter, darunter SMK, LinTech, connectBlue, bieten Module mit dieser Funktionalität an, wobei die Wahl aufgrund der sehr guten Abdeckung aller Anforderungen auf ein connectBlue Modul viel. Durch den Einsatz eines OEM Moduls können auch die Anforderungen an Größe und Gewicht erfüllt werden. Zur Auswahl standen somit die Module cB-OEMSPA33i und cB-OEMSPA13i, wobei es sich beim 13i um ein Gerät der Klasse 2 (Reichweite lt. Hersteller 10 - 30m) mit einer Größe von 23mm x 36mm x 5mm handelt. Der Adapter 33i ist ein Klasse 1 Modul (Reichweite lt. Hersteller > 100m) mit einer Größe von 40mm x 42mm x 5mm. Beim Stromverbrauch ist Herstellerangaben zu Folge kaum ein Unterschied zwischen den beiden Modulen. Da speziell in Gebäuden die maximale Reichweite stark eingeschränkt ist wurde dem OEMSPA33i (Abbildung 27) trotz etwas größerer Bauform der Vorzug gegeben.

Die wichtigsten Eigenschaften [7] des gewählten Moduls sind:

- Versorgungsspannung 3,3 – 6 V
- Leistungsaufnahme: abhängig von Zustand, durchschnittlich 78mA beim senden, abhängig von der Datenrate, den Daten, usw., siehe [6]
- Datenrate bis zu 921,6 kBits/sec
- Logische Eingangspiegel: UART, RS232
- Reichweite > 100 m
- Abmessungen 40mm x 42mm x 5mm
- Interne Antenne

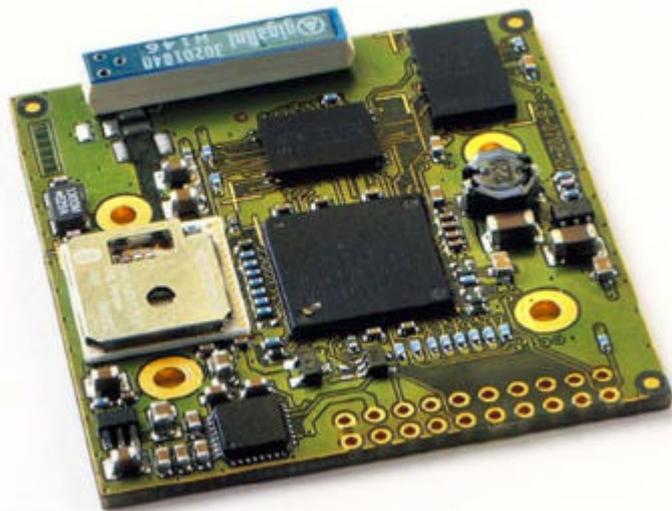


Abbildung 27: connectBlue Serial Port Adapter 33i

Das Bluetooth Modul hat zusätzlich den Vorteil, dass es sehr einfach mittels dazugehöriger Platine oder auch via Bluetooth konfiguriert werden kann. Hierbei können vier wesentliche Punkte eingestellt werden:

- Eigenschaften der seriellen Schnittstelle (Baudrate, Datenbits, ...)

- Bluetooth Einstellungen (Sicherheitskriterien, Modulname, ...)
- Verbindungseinstellungen (Client/Server)
- Authentifizierung von Geräten (Pairing)

Die Eigenschaften der seriellen Schnittstelle (Baudrate 115,2 kBits/s, acht Datenbits, ein Paritybit, ein Stopbit, keine Flusskontrolle) sind vom Sensor abhängig. Diese Konfiguration des Bluetooth Moduls ist notwendig um bei bestehender Bluetooth-Verbindung eine Verbindung mit dem Sensor herstellen zu können.

5.4 Weitere Hardware

Nach der Auswahl der Sensorik und der Hardware für die Datenübertragung erfolgte die Dimensionierung und Auswahl der restlichen Hardware die für den Betrieb notwendig ist. Im Wesentlichen geht es um die Akkumulatoren und die dazugehörige Spannungsversorgung der beiden bereits ausgewählten Komponenten.

5.4.1 Akkumulatoren

Die Anforderungen an die Akkus sind durch die Problemstellung festgelegt. Die unterste Grenze für die Laufzeit ist mit einer Stunde angegeben, wobei jedoch ein autonomer Betrieb von zwei Stunden wünschenswert ist, die weiteren Anforderungen sind wiederum Größe und Gewicht. Aufgrund der Sensorik muss die Spannung mindestens 6V betragen. Da bereits fertige Akkupacks die gewünschten Eigenschaften nicht ausreichend gut erfüllten, wurden diese selbst hergestellt (Abbildung 28). Hierzu wurden SANYO Nickel Metallhydrid Akkus des Typs HF-B1U mit Lötflächen gewählt. Diese Akkus bieten eine besonders hohe Energiedichte. Bei der Nennspannung von 1,2V pro Zelle und einer Kapazität von typischerweise 860mAh beträgt das Gewicht 18g. Durch die Maße von 48mm x 6,1mm x 17mm pro Zelle ergeben sich für einen Akkupack mit fünf Zellen ein Volumen von rund 25cm³ bei einem Gewicht von 90g und einer Nennspannung von 6V, wobei die tatsächliche Spannung im geladenen Zustand bei rund 7 - 7,2V liegt, diese fällt im Betrieb jedoch relativ schnell ab.

Die Akkus sind somit der für das Gesamtgewicht der entscheidende Faktor (Sensor 25g), jedoch sind die 90g (bzw. das Gesamtgewicht mit allen Kabeln usw. rund 135g) durchaus in einem Bereich bei dem keine tatsächliche Beeinträchtigung der Bewegungsfreiheit der Hände eintritt.



Abbildung 28: Akkupack für die Hardware

Bei den Steckern der Akkupacks gibt es zwei verschiedene Versionen, siehe Abbildung 29 (es existieren zwei komplette Prototypen, jeweils mit Reserveakkus). Bei der ersten Version wurde primär auf die Größe des Steckers geachtet, jedoch mit dem Nachteil das dieser nicht verpolungssicher ist und somit die Aufmerksamkeit des Anwenders erfordert, dies wurde bei der zweiten Version geändert, indem ein anderes, größeres, Steckersystem verwendet wird.

Die Hardware wurden generell möglichst modular (z.B. durch Steckverbindungen) aufgebaut um während der Testphasen die einzelnen Teile einfach auswechseln zu können und um sich möglichst flexibel den unterschiedlichen Platzangebot in den verschiedenen Objekten anzupassen, diese Artefakte werden in Abschnitt 5.10 näher erläutert.



Abbildung 29: Stecker der Akkupacks

5.4.2 Spannungsregler

In der Schaltung werden zwei unterschiedliche Spannungen benötigt. Der Sensor wird normalerweise mit 6V +/-10% betrieben, jedoch ist laut Hersteller auch ein Betrieb mit einer Spannung von bis zu 9V möglich, hierbei kann es aufgrund der zusätzlichen Leistung allerdings zu geringfügigen Langzeitdrifts der Sensorwerte kommen.

Da die Spannung der Akkus nach kurzer Zeit relativ stark vom Maximalwert (bis zu 7,2V) abfällt wurde keine weitere Maßnahme getroffen um für den Sensor eine eigene Spannungsregelung vorzunehmen.

Das Bluetoothmodul kann mit einer Spannung von 3,3V bis maximal 6V betrieben werden, somit ist kein direkter Betrieb mit den Akkus möglich. Es wurde ein Festspannungsregler (LM2940T-5.0) mit einer Ausgangsspannung von 5V gewählt. Die Beschaltung erfolgt nach der Referenzschaltung in [35] und ermöglicht somit einen spannungsschwankungsfreien Betrieb.

Als Kabel für den Anschluss des Sensors an das Bluetoothmodul wurde ein Flachbandkabel von Samtec (Bezeichnung: TCSD-10-D-02.00-01) gewählt, dieses Kabel wurde jedoch so geändert, dass die gesamte Spannungsregelung (Festspannungsregler und Kondensatoren) direkt mit dem Kabel verbunden ist. Es wurde ein Kabel mit Steckverbindungen gewählt um den Sensor bzw. das Funkmodul auch weiterhin in anderen Anwendungen verwenden zu können bzw. um Änderungen ohne großen Aufwand vorzunehmen.

5.5 Systemüberblick über die Hardware

Die Anforderungen, die aufgrund des Wunsches nach einem autonomen Betrieb gestellt wurden, können mit den gewählten Komponenten erfüllt werden. So liegt etwa die Betriebsdauer bei zweieinhalb bis über vier Stunden, gefordert war eine Dauer von zumindest einer Stunde. Bei Reichweitenmessungen konnten unter idealen Bedingungen (im Freien) mit einem ACER BT500 Bluetooth Dongle als Gegenstelle Entfernungen von 85-90m überbrückt werden. In Gebäuden bricht, je nach Bauweise, die Verbindung auch über eine Distanz von mehreren Räumen nicht ab.

Der Gesamtpreis pro System beträgt rund 1600 Euro, wobei der Großteil, wie bereits erwähnt, durch den Sensor verursacht wird, das Bluetoothmodul (rund 160 Euro) und die Akkus (45 Euro) spielen eine geringe Rolle.

In Abbildung 30 ist die gesamte Hardware im Größenvergleich zu einer Zigarettenschachtel zu sehen.



Abbildung 30: Größenvergleich der Hardware

Bei den Versuchen konnte eine Verfälschung der Sensorwerte durch größere metallische Gegenstände in Sensornähe festgestellt werden. Dies äußert sich beispielsweise wenn sich ein Akkupack sehr nahe am Sensor befindet. Bei einem Abstand der Akkumulatoren zum Sensor in der Größenordnung von 3-4 cm ist die Beeinflussung der Sensorwerte kaum mehr festzustellen und stört somit die Bewegungserkennung nicht. Die Verfälschung der Sensorwerte macht sich etwa bemerkbar, wenn der Sensor aus einer Startorientierung gedreht wird und danach wieder die ursprüngliche Orientierung annimmt, die Sensorwerte können nun jedoch teilweise sehr stark von den ursprünglichen Werten abweichen, in extremen Fällen (große metallische Gegenstände direkt am Sensor) ändern sich sogar die Sensorwerte ohne das eine Bewegung stattgefunden hat. Dieses Verhalten tritt auch wenige Minuten bevor die Akkus leer werden und die Verbindung abbricht auf.

Durch die Abhängigkeit zwischen der Sensorik und metallischen Gegenständen können sich gegebenenfalls auch Einschränkungen für den Betrieb ergeben. Wenn es beispielsweise nur möglich ist den Sensor ganz am Rand eines Artefaktes einzubauen und somit der Abstand zu metallischen Gegenständen in der Umgebung (z.B. metallische Tischoberfläche) sehr gering ist, so können diese Verfälschungen entstehen, auch bei der Integration der Hardware in metallische Gegenstände können Probleme auftreten. Die tatsächlichen Auswirkungen sind jedoch von Fall zu Fall zu erheben.

Bei unterschiedlichen Anwendungen sind verschiedene Anforderungen an die Genauigkeit gegeben, die Möglichkeiten der Sensorik werden jedoch kaum ausgeschöpft, deswegen kann bei Bedarf auch eine Beeinflussung durch metallische Gegenstände in Kauf genommen werden, da die Qualität der Sensorwerte trotz Verfälschung für die entsprechende Anwendung noch immer ausreicht.

5.5.1 Blockschaltbild

Der strukturelle Aufbau der Hardware mit den vier Hauptkomponenten Spannungsregler, Akkus, Sensorik und Funkmodul und deren Zusammenspiel ist in Abbildung 31 zu sehen.

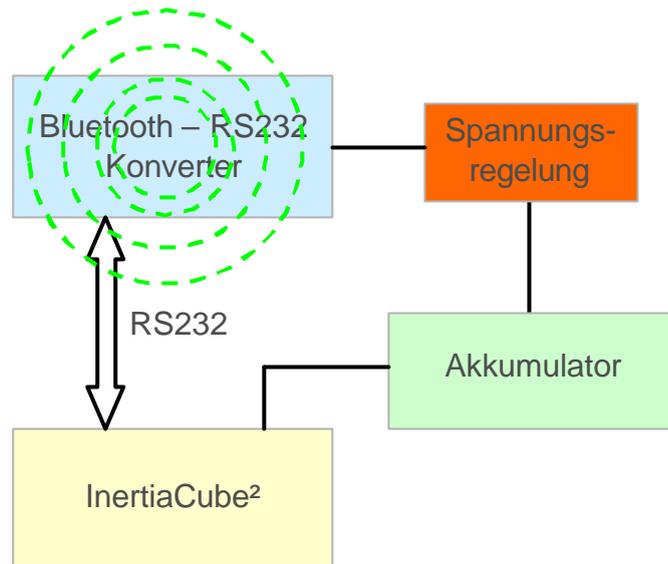


Abbildung 31: Schematischer Aufbau des Hardware Prototypen

5.6 Anforderungen an die Software

Die Software soll die Verwendung der Bewegungserkennung unabhängig von der darunter liegenden Hardware möglichst einfach gestalten. Durch die Abstrahierung von der Hardware soll es auch reinen Softwareentwicklern ermöglicht werden, ohne spezielles Wissen über die Hardware und die verwendete Sensortechnologie, die Bewegungserkennungssoftware zu verwenden. Da die Gesten unmittelbar nachdem sie ausgeführt wurden erkannt werden sollen, ist es nicht möglich die Sensorwerte im Nachhinein auf entsprechend optimierten Systemen zu analysieren und die einzelnen Gesten auszuwerten. Weiters soll die Möglichkeit bestehen die Gesten, welche überprüft werden sollen, zur Laufzeit zu verwalten, also hinzuzufügen und auch zu entfernen.

Es sollen bereits Gesten für verschiedene geometrische Formen zur Verfügung gestellt werden um eine schnelle Entwicklung zu erlauben. Zusätzlich sollen weitere beliebige Gesten definiert werden können die bei der Gestenerkennung verwendet werden können. Neben dem Zugriff auf die erkannten Gesten, also auf eine stark abstrahierte Ebene der Sensordaten, ist es auch erforderlich, dass die Sensordaten zur Verfügung gestellt werden.

Obwohl die Auswertung der Sensordaten auf einem verhältnismäßig leistungsstarken Notebook erfolgt, muss Ressourcen schonend vorgegangen werden, da einerseits die Bewegungserkennung nur ein Teil der Anwendung ist und etwa bei einer Aktualisierungsrate eines Sensors von 180Hz alle rund 5ms ein neuer Sensorwert vorhanden ist für den die Bewegungserkennung, abhängig von der Programmierung auch nur ein Teil davon, durchgeführt werden muss.

Die Erkennungsrate der Gesten ist nicht das primäre Ziel, entsprechend sind die Möglichkeiten auf diesem Gebiet noch nicht vollständig ausgereizt. Aufgrund der Struktur ist es jedoch möglich die entsprechenden Algorithmen einfach auszutauschen und somit die Erkennungsrate zu steigern.

Als Entwicklungssprache wurde Java gewählt. Hierbei ist neben der Unterstützung von verschiedensten Plattformen auch eine mögliche spätere Ausführung der Software auf Geräten die Java unterstützen ein zentraler Punkt. Wobei die derzeitige Version der Software auf Geräten die lediglich die J2ME (Java 2 Micro Edition) unterstützen nicht verwendet werden kann, da Teile von Java verwendet werden, die in J2ME nicht vorhanden sind.

5.7 Software für unterschiedliche Sensorik

Da die Bewegungserkennung mit unterschiedlichen Sensoren funktionieren soll, ist es erforderlich eine einheitliche Schnittstelle in Form eines Interfaces zu erstellen. Dies ist die einzige Schnittstelle zur Kommunikation mit der Hardware, wobei diese vom Anwendungsprogrammierer verborgen bleibt und nur im Fall eines neuen Sensors von Interesse ist, da in diesen Fall eine Klasse geschrieben werden muss, die dieses Interface implementiert. Das Interface `OrientationSensor` umfasst insgesamt vier Methoden, diese sind:

- `boolean` `openSensor()`
- `boolean` `closeSensor()`
- `OrientationSensorData` `getSensorData()`
- `OrientationSensorInfo` `getSensorInfo()`

Die Methoden `openSensor()` und `closeSensor()` dienen zur der Initialisierung der Sensoren. Hierbei müssen die Verbindung zum Sensor hergestellt, und sonstige notwendige Konfigurationen die den Sensor oder die gesamte Hardware betreffen durchgeführt werden. Nach dem Aufruf von `openSensor()` müssen die Informationen über den Sensor und die aktuellen Sensorwerte abgefragt werden können.

Der Datentyp `OrientationSensorInfo` enthält alle wesentlichen Informationen zum Sensor, diese Daten werden normalerweise selten abgefragt, `OrientationSensorData` hingegen enthält Sensordaten, dies bedeutet, dass `getSensorData()` ständig aufgerufen wird wenn für die Bewegungserkennung die aktuellen Sensorwerte benötigt werden, dies kann etwa beim InertiaCube² mit bis zu 180 Hz erfolgen bzw. kann der Aufruf öfters erfolgen jedoch werden vom Sensor nur bis zu 180 Sensorwerte in der Sekunde zur Verfügung gestellt. Beim schließen mittels `closeSensor()` sollen die verwendeten Ressourcen wieder freigegeben werden.

`OrientationSensorInfo` dient der Beschreibung des Sensors und somit müssen die einzelnen Felder keine bestimmten Kriterien erfüllen. Bei `OrientationSensorData` hingegen müssen die Felder `yaw` und `roll` Werte im Bereich $[0..360[$ und `pitch` einen Wert im Bereich von $[0..180[$ annehmen. Dass der Wertebereich von `pitch` ein anderer ist hängt mit einer Eigenheit des InertiaCube² von InterSense zusammen, dieses Verhalten müssen somit auch alle anderen Sensoren die verwendet werden aufweisen. Sobald der `pitch`-Wert (Drehung um die `y`-Achse) den Wert 180 erreicht springen die beiden anderen Werte (`yaw` und `roll`) um 180. Zusätzlich erhöht sich der `yaw`-Wert bei weiterer Drehung um die `y`-Achse nicht, sondern verringert sich, d.h. bei einem Startwert von $\langle \text{yaw}, \text{pitch}, \text{roll} \rangle = \langle 120, 150, 340 \rangle$ und einer Drehung von 50 um die `y`-Achse ergeben sich somit die Werte $\langle 300, 160, 160 \rangle$. Die Werte des Sensors sind jeweils in Grad angegeben, eine komplette Drehung um eine Achse entspricht somit 360° , eine Drehung um $p/2$ Radianen entspricht in diesen Wertebereich 90° .

Diese Vorgehensweise widerspricht zwar der Forderung nach einer möglichst Hardware-unabhängigen Implementierung, jedoch ist es erforderlich die Software an die minimalen Gegebenheiten möglichst aller Sensoren dieser Klasse anzupassen. Da die Sensorwerte von anderen Sensoren bei der Implementierung des Interfaces leicht an diesen Umstand angepasst werden können, wurde dieser Weg gewählt. Als Erstes wurde eine Anpassung des `pitch` Wertes des InertiaCube² auf den Bereich von $[0..360[$ angestrebt, dies konnte jedoch nicht realisiert werden, da es sich bei diesen Sensor um keinen idealen Sprung handelt der genau beim Überschreiten des Wertes 180 eintritt, sondern um eine Erscheinung die langsam vonstatten geht, so ändern sich die Werte `yaw` und `roll` bereits leicht sobald `pitch` einen Wert von rund 165 annimmt und nimmt zu, desto weiter sich `pitch` den Wert 180 nähert. Diese

Tatsache macht es unmöglich, eindeutig zu erkennen ob es sich um eine Bewegung oder ob es sich um diesen Sprung handelt. Diese Eindeutigkeit wäre jedoch für die Verwendung für die Bewegungserkennung erforderlich.

Diese Eigenschaft des Sensors hat in weiteren Teilen der Software auch die Auswirkung, dass Bewegungen die so einen Sprung beinhalten im Bereich der Sprungstelle nur eingeschränkt festgelegt werden können, da beispielsweise der yaw-Wert bei einem vorgegebenen pitch-Wert trotz beinahe identischer Ausführung der Bewegung sehr stark abweichen kann. Um diese Fehlerquelle zu umgehen bzw. zu minimieren werden in so einem Fall die Werte yaw und roll bei der Bewegungserkennung nicht betrachtet. Speziell wenn die Bewegung nach einem Sprung noch nicht beendet ist ergibt sich dadurch keine wesentliche Verschlechterung der Erkennungsrate.

Dieses Verhalten des InertiaCube² wird im Datenblatt des Sensors nicht angeführt, tritt jedoch bei allen drei getesteten Sensoren auf.

5.7.1 Einbinden des InertiaCube²

Im Lieferumfang des InertiaCube² befindet sich bereits eine Software, welche die Sensordaten auf die zuvor geforderten Bereiche abbildet, es ist somit keine Umrechnung von Abtastwerten oder ähnlichem auf die Orientierung des Sensors in Grad erforderlich. Da die Bewegungserkennung in Java erfolgt, die Sensordaten jedoch mittels einer DLL (Dynamic Link Library) zur Verfügung gestellt werden ist ein Zugriff auf die Methoden zur Kommunikation mit dem Sensor mittels JNI (Java Native Interface) notwendig. Die in [20] dokumentierten und für diese Anwendung relevanten Methoden sind somit auch von Java aus nutzbar und werden in der Implementierung des Interfaces `OrientationSensor` für den InertiaCube² aufgerufen.

5.8 Einbindung der Gestenerkennung in ein Programm

Das umfangreiche API ist mittels Javadoc dokumentiert, einige wesentliche Möglichkeiten und die dahinter liegenden Konzepte werden hier herausgegriffen und erläutert. Weiters wird gezeigt, dass die Integration der Bewegungserkennung in ein Programm ohne großen Aufwand möglich ist.

5.8.1 „Hello world“ für Gestenerkennung

Das Programm `HelloWorld` zeigt alle für die Bewegungserkennung minimal notwendigen Anweisungen.

```
import trackingFramework.*;

public class HelloWorld implements TrackingEventListener {
    private static HelloWorld helloWorld;

    public void trackingEventPerformed (TrackingEvent e) {
        System.out.println("Bewegung mit der ID: " + e.getID() +
            " wurde erkannt: " + e.getMsg());
    }

    public static void main(String[] args) {
        helloWorld = new HelloWorld ();
        Framework.addListener((TrackingEventListener) helloWorld,
            Framework.ORIENTATION_CHANGE);

        if (Framework.initSensor()) {
            System.out.println("Initialisierung des Sensors erfolgreich");
        }
    }
}
```

```

    try {
        Thread.sleep(60000);
    } catch (InterruptedException e) {
        System.out.println("InterruptedException" + e);
    }
    Framework.closeSensor();
}
}

```

Die Funktionalität ist einfach beschrieben. Nach der Initialisierung des Sensors wird eine Minute auf eine Änderung des Sensorwertes gewartet, wenn dies eintritt wird eine entsprechende Statusmeldung ausgegeben.

Folgende Punkte sind für die Bewegungserkennung erforderlich:

- `trackingFramework` muss importiert werden
- Implementieren des Interfaces `TrackingEventListener`
- Registrieren der gewünschten Gesten
- Initialisieren des Sensors
- Beim beenden schließen des Sensors

Das Interface `trackingEventListener` enthält die Methode `trackingEventPerformed(TrackingEvent e)`, diese wird aufgerufen sobald eine Geste erkannt wird, näheres ist hierzu in Kapitel 5.8.3 zu finden. Damit eine Geste erkannt werden kann, müssen die gewünschten Gesten zuvor am Framework zur Bewegungserkennung registriert werden, wobei entweder bereits definierte Gesten verwendet werden können, oder es werden Gesten durch den Anwendungsprogrammierer frei definiert. Um nun die Sensorwerte verwenden zu können und die Bewegungserkennung zu nutzen muss noch der Sensor initialisiert werden, am Ende muss dieser wieder geschlossen werden.

5.8.2 Konfigurationsmöglichkeiten

Die Möglichkeiten zur Konfiguration beziehen sich hier nur auf zentrale Eigenschaften der Bewegungserkennung und nicht auf angrenzende Teile wie die Visualisierung der Bewegungen.

Es stehen mehrere Methoden zur Verwaltung eines Offsets zur Verfügung. Diese werden nicht nur benötigt wenn der Sensor in einer bestimmten Lage montiert wird und somit andere Werte liefert, sondern auch für die unterschiedlichen Bewegungen, da es hier die Unterscheidungsmöglichkeit zwischen Bewegungen mit fixen und Bewegungen mit variablen Startpunkt gibt. Für die Gesten mit fixem Startpunkt kann mithilfe des Offsets der Sensorwert an die Anforderungen der Anwendung bzw. der Geste angepasst werden.

Ein weiterer wichtiger Wert kann mittels `setDelta(float delta)` eingestellt werden. Dieser Wert ist die minimale Änderung seit der Abfrage des letzten Sensorwertes die eintreten muss, um als neuer Wert erkannt und gewertet zu werden. Die Idee die dahinter steckt ist es das kleine, für den Menschen unbewusste, Änderungen des Sensorwertes durch minimale Bewegungen oder auch durch Ungenauigkeit des Sensors keine ständigen Auswirkungen haben sollen, auch wenn durch solch eine kleine Änderung der Delta-Wert überschritten werden kann. Weiters ist es ein effektives Mittel die Anzahl der Bewegungserkennungsberechnungen gering zu halten. Es ist beispielsweise für den Großteil der derzeitigen Anwendungen und auch für die Bewegungserkennung ausreichend diesen Wert auf 5° zu stellen. Die Überprüfung ob eine Geste ausgeführt wird erfolgt erst wenn sich zumindest einer der drei Werte (yaw, pitch, roll) seit der letzten Berechnung um zumindest diesen Wert (5°) geändert hat.

Diese Einstellung ist eng mit einem weiteren Parameter, der Aktualisierungsrate, verbunden. Mittels `setSensorUpdateRate(int sensorUpdateRate)` kann festgelegt werden, wie oft der aktuelle Wert des Sensors für die Bewegungserkennung abgefragt wird. Änderungen dieses Wertes sollten jedoch immer mit der maximalen Aktualisierungsrate des aktuell verwendeten Sensors abgestimmt werden, da eine höhere als durch den Sensor unterstützte Aktualisierungsrate keine zusätzliche Information über die momentane Orientierung bringt, jedoch ein Mehraufwand durch die zusätzliche Kommunikation mit dem Sensor und natürlich auch in der Berechnung verursacht.

5.8.3 Konzept der Gesten

Die Idee hinter dem hier verwendeten Konzept zur Mitteilung der erkannten Gesten ist an die im Umgang mit der Maus oder konventionellen GUIs bekannten Events angelehnt. Hierzu muss das Interface `trackingEventListener`, also die Methode `trackingEventPerformed (TrackingEvent e)` implementiert werden. Diese Methode wird aufgerufen sobald eine Geste erkannt wurde, wenn mehrere Gesten gleichzeitig, also durch eine Änderung des Sensorwertes, erkannt wurden wird die Methode für jede Geste einmal mit dem entsprechenden `TrackingEvent` aufgerufen. Das `TrackingEvent` enthält neben einer kurzen Beschreibung der Geste auch eine eindeutige ID, mit der jede Geste identifiziert werden kann. Anhand dieser ID kann entschieden werden welche Aktion ausgeführt werden soll.

Der Vorteil in diesem Event-Konzept liegt in der einfachen Handhabung. Die Ereignisbehandlung funktioniert nach dem Delegationsmodell, hierbei registrieren sich Objekte direkt bei der Quelle eines bestimmten Ereignisses, daraufhin werden diese Objekte automatisch über das Auftreten des Ereignisses informiert. Nach der Registrierung der Gesten sind bis zum Eintreten eines Events, also dem Erkennen einer Geste, keine weiteren Schritte mehr erforderlich.

5.8.4 Unterteilung der Gesten

In der Implementierung wird zwischen drei unterschiedlichen Arten von Gesten unterschieden, dies hat primär Performancegründe, es gibt jedoch auch noch weitere Eigenschaften die nur für bestimmte Teile der Gesten zutreffen und somit aufgrund dieser Unterteilung einfacher verwaltet werden können. Diese drei Arten von Gesten sind *Zustand*, *Zustandsüberführung* und *Bewegung*.

Zustand

Ein Zustand ist die einfachste Art, hierbei handelt es sich lediglich um eine bestimmte Orientierung, also ein Tripel der Form `<yaw, pitch, roll>`. Für jeden der drei Werte kann ein Delta-Bereich angegeben werden in dem sich der aktuelle Sensorwert befinden muss, damit dieser Zustand eintritt, es kann hierbei von einer statischen Geste gesprochen werden. Weiters kann hier die Betrachtung von drei auf zwei oder nur einen relevanten Wert reduziert werden, somit können ein-, zwei- oder dreidimensionale Grenzen vorgegeben werden in denen die Sensorwerte liegen müssen. Sobald der Sensorwert in diesen Bereich liegt wird ein Event ausgelöst, dieses Event wird erst dann erneut ausgelöst, wenn die Bedingungen für diesen Zustand zwischendurch nicht mehr erfüllt waren, dies hat den Grund, dass nicht bei jeder Änderung der Sensorwerte ein Event ausgelöst werden soll. Die Erkennungsroutine für diese Gruppe ist einfach und sehr laufzeiteffizient zu realisieren.

Zustandsüberführung

Zustandsüberführungen basieren auf Zuständen und nutzen den Vorteil der einfachen Berechnung und die Möglichkeit, dass bestimmte Punkte für mehrere Bewegungen von

Interesse sind. Hierzu bildet eine Reihe von Zuständen die nacheinander eintreten eine Zustandsüberführung. Die Zustände müssen in der Reihe in der sie angegeben sind eintreten damit eine Erkennung der Zustandsüberführung möglich ist. Zusätzlich gibt es bei dieser Gruppe noch ein paar Parameter die diese Gesten näher beschreiben, so ist es beispielsweise möglich eine maximale Dauer zwischen der Erkennung des ersten und des letzten Zustandes der Folge festzulegen. Wenn ein detaillierteres Wissen über alle Zustände die überprüft werden vorhanden ist, kann die Anzahl der Zustände festgelegt die zwischen zwei Zuständen einer Zustandsüberführung erkannt werden dürfen, hierbei muss jedoch darauf geachtet werden, dass jederzeit neue Gesten definiert werden können und somit dieses Ergebnis beeinflussen können. Gesten wie das Umlegen einer Pyramide oder auch eine bestimmte Reihenfolge von Kippbewegungen im Umgang mit einem Würfel, indem die bereits vordefinierten Zustände für Würfel verwendet werden, können mit Zustandsüberführungen einfach modelliert werden. Neben der sehr einfachen Realisierung von Gesten, die gewisse Ähnlichkeiten mit endlichen Automaten haben, bringt dieser Ansatz bei einer sehr großen Anzahl von Zuständen bzw. wenn mehrere Zustände von unterschiedlichen Zustandsüberführungen verwendet werden einen Performancevorteil.

Bewegung

Bei den Bewegungen gibt es noch etwas umfangreichere Möglichkeiten bei der Definition der Gesten. Prinzipiell besteht eine Bewegung aus einer Folge von Tripel mit einem dazugehörigen Delta-Wert, somit ergibt sich eine Art Schlauch indem sich die Werte des Sensors befinden müssen damit eine Bewegung erkannt werden kann. Aufgrund sehr schneller Änderungen des Sensorwertes (z.B. durch schnelle Handbewegungen) ist es möglich, dass mehrere Punkte dieser Punktfolge übersprungen werden. Weiters gibt es die Möglichkeit zusätzliche Angaben zur Geschwindigkeit der Bewegung zu machen, indem eine minimale und eine maximale Dauer der Geste angegeben werden kann. Ein wichtiger Unterschied zu den Zustandsüberführungen sind die fixen bzw. variablen Startpunkte der Gesten, diese Eigenschaft kann bei der Definition der Bewegung festgelegt werden. Bei den fixen Startpunkten muss der Sensor eine bestimmte Lage, abhängig von der Bewegung, einnehmen. Ein fixer Startpunkt ist somit mit einem Zustand zu vergleichen. Erst wenn dieser eingetreten ist werden die folgenden Punkte der Bewegungsdefinition überprüft. Die variablen Startpunkte können zur Laufzeit festgelegt werden, dies erfolgt in der derzeitigen Implementierung etwa durch das Überschreiten einer bestimmten Rotationsgeschwindigkeit. Wird dieser Schwellwert überschritten, so wird der Startpunkt der Bewegung auf den aktuellen Wert des Sensors festgelegt. Dies hat nun etwa den Vorteil das eine Geste wie das Drehen um 90° von einer beliebigen Startposition ausgehend erkannt werden kann, wobei die Bewegungen natürlich auch entsprechend komplexer sein können. Auch bei der Festlegung des Startpunktes sind andere Kriterien möglich.

Bewegungen die bereits zu einem Teil erkannt wurden und bei denen sich dann der Startteil der Geste innerhalb einer Geste wiederholt, können in der aktuellen Version unter bestimmten Voraussetzungen nicht erkannt werden, dies hängt jedoch nicht mit der Sensorik sondern lediglich mit dem verwendeten Algorithmus zusammen. Dieser Fall tritt jedoch selten auf und kann zusätzlich durch die Modellierung der Gesten umgangen werden, notfalls muss eine Geste auf zwei Teilgesten aufgeteilt werden.

Der derzeit verwendete Algorithmus zur Bewegungserkennung kann aufgrund der modularen Programmierung jederzeit durch ein performanteres Verfahren oder eines mit einer höheren Erkennungsrate ausgetauscht werden.

5.8.5 Hinzufügen und entfernen von Gesten

Es besteht die Möglichkeit jederzeit Gesten zu den zu überprüfenden Gesten hinzuzufügen und auch wieder zu entfernen. Somit erfolgt die Bewegungserkennung nur für Gesten welche

aktuell auch wirklich benötigt werden. Da auch die vom Entwickler definierten Gesten in einem Gestenvorrat gespeichert bleiben kann jederzeit ohne großen Aufwand eine Geste wieder zu den zu überprüfenden Gesten hinzugefügt werden, hierzu muss nur die ID der Geste angegeben werden.

Der Anwender kann sehr einfach neue Gesten erstellen, hierzu müssen lediglich die einzelnen Punkte, bzw. bei einem Zustand nur ein Punkt, bei einem entsprechenden Objekt hinzugefügt werden. Dieses Objekt muss danach mittels add-Methode beim Framework registriert werden, hierbei wird eine eindeutige ID zurückgegeben. Mithilfe dieser ID kann während des Ausführens des Programms auf die Geste zugegriffen, und die Eigenschaften können bei Bedarf geändert werden. Die IDs sind in insgesamt sechs Bereiche unterteilt, jeweils drei betreffen die bereits vordefinierten und die vom Anwendungsentwickler definierten Gesten, wobei die drei Gruppen für Zustände, Zustandsüberführungen und Bewegungen sind. Durch diese Einteilung beinhaltet die ID bereits eine gewisse Information über die Art der Geste.

Durch das flexible Anlegen neuer Gesten ist es auch möglich die Gesten durch den tatsächlichen Benutzer des TUIs festlegen zu lassen. Hierzu muss der Benutzer die gewünschte Geste ausführen, hierbei werden die Sensorwerte aufgezeichnet. Aus diesen Daten wird eine Geste angelegt und überprüft ob die Geste generell erkannt werden kann, so sind für eine Bewegung z.B. zumindest zwei Punkte erforderlich. Bei Bedarf kann eine zusätzliche Überprüfung mit den bereits verwendeten Gesten auf eine eventuell erforderliche Eindeutigkeit erfolgen. Danach kann die durch den Benutzer vorgezeigte Bewegung registriert und sofort verwendet werden. Der Benutzer kann somit bei Bedarf die Gesten selbst festlegen und ist bei entsprechender Entwicklung der Anwendung nicht auf die vorgegebenen Gesten angewiesen.

5.8.6 Log- und Visualisierungsmöglichkeiten

Neben einer eigenen umfangreichen Log-Funktion gibt es mehrere Möglichkeiten zur Visualisierung der Bewegungen.

Eine Darstellung von Bewegungen kann in mehreren Fällen von Interesse sein. Eine Möglichkeit ist eine eigene Anwendung welche die Bewegungen des Benutzers auch visuell darstellt. Dies kann zum Vorzeigen der prinzipiellen Möglichkeiten des Sensors bzw. der gesamten Hardware genutzt werden. Eine weitere Anwendung könnte ein Trainer für bereits bestehende Bewegungen sein. Der Verlauf der zu trainierenden Bewegung wird gezeichnet, zusätzlich kann auch noch der Delta-Bereich transparent dargestellt werden, somit ergibt sich eine Art Schlauch in dem sich der Sensorwert ständig befinden muss. Der Benutzer sieht folglich immer welche Auswirkungen die Bewegungen haben und wie weit er von den Grenzen der Geste entfernt ist, sobald diese Grenze überschritten ist kann dies entsprechend grafisch dargestellt werden. Eine visuelle Trainingsfunktion könnte speziell bei komplexeren Bewegungen von Interesse sein, wobei die derzeit verwendeten Bewegungen auch ohne Trainingsfunktion leicht durchzuführen sind. Diese Trainingsfunktion wurde somit noch nicht implementiert, jedoch könnte dies basierend auf den Visualisierungsmöglichkeiten einfach erfolgen.

Es existieren zwei grundlegend unterschiedliche Möglichkeiten zur Bewegungsdarstellung. In einer wird eine VRML (Virtual Reality Modeling Language) Datei erstellt, diese beinhaltet einen Linienzug aller Punkte die vom Benutzer zur Darstellung hinzugefügt wurden und ist für die nachträgliche Betrachtung in einem entsprechenden Viewer gedacht. Der zeitliche Verlauf kann durch Würfel auf diesen Linienzug, welche abhängig von der Zeit die Farbe ändern, dargestellt werden.

Bei der zweiten Möglichkeit handelt es sich um die dynamische Repräsentation der Sensorwerte mittels Java3D. Hierbei kann neben den Sensorwerten in Echtzeit auch noch der

Zeitverlauf dargestellt werden. Eine beliebige Anzahl von Würfeln stellen die Sensorwerte für einen frei wählbaren Zeitraum dar. Die Position der Würfel ist vom Wert des Sensors abhängig, die Transparenz der Würfel hingegen von der Zeit die vergangen ist seit dem der Sensor diesen Wert angenommen hat. Durch den Abstand der Würfel wird indirekt die Geschwindigkeit bei der Bewegung visualisiert. Beispiele für die Anwendung beider Methoden befinden sich in Abschnitt 6.1.

5.9 Überblick über das gesamte System

Neben den bisher beschriebenen Hard- und Softwarekomponenten sind für den Betrieb noch weitere Teile erforderlich. Komponenten die nicht für die eigentliche Bewegungserkennung oder für die Ausführung mehrere Anwendungen erforderlich sind werden bei den entsprechenden Szenarien in Abschnitt 6 erwähnt.

5.9.1 Verwendete Hardware

Zur Auswertung der Sensordaten und als Plattform für die jeweiligen Anwendungen wurden zwei verschiedene mobile Geräte verwendet. Zum einen ein Laptop (P3 850MHz, 256MB RAM, Windows XP Professionell) und zum anderen ein Tablet PC (1GHz, 512MB RAM, Windows XP Tablet PC Edition). Bei den Anwendungen wurden bis zu 25 verschiedene Gesten gleichzeitig verwendet, für diese geringe Anzahl von Bewegungen reicht die vorhandene Rechenleistung trotz einer Aktualisierungsrate von 180 Hz leicht aus. Zukünftig müssten somit auch kleinere mobile Geräte mit geringeren Ressourcen für diese Auswertung genützt werden können.

Weiters wurde bei allen Anwendungen ein ACER Bluetooth Dongle (BT500) für die Datenübertragung vom Sensor zur Auswertestation (Laptop bzw. Tablet PC) verwendet. Die einzige Anforderung an den Bluetooth Dongle ist, dass er das Bluetooth Profil für den seriellen Port unterstützt. Für Anwendungen die mehrere Sensoren benötigen muss ein entsprechender Dongle verwendet werden der es ermöglicht mehrere serielle Verbindungen gleichzeitig aufrecht zu erhalten.

Damit ein erfolgreicher Aufbau einer Verbindung zum Sensor möglich ist muss die Hardware natürlich mit Strom versorgt werden und das Bluetooth Modul des Sensors muss einmalig entsprechend konfiguriert werden.

5.9.2 Notwendige Software

Neben dem erstellten Framework ist noch weitere Software von Drittherstellern erforderlich damit die Anwendungen ausgeführt werden können.

Einmalig wird die Software zur Konfiguration des connectBlue Bluetooth Moduls benötigt, dieses Programm ist im Lieferumfang des Moduls enthalten. Wenn die Konfiguration nicht über Bluetooth durchgeführt wird, so kann dies über eine zusätzliche Platine erfolgen die mittels RS232 mit dem PC verbunden wird.

Das Framework für die Bewegungserkennung benötigt eine JVM (Java Virtual Machine) ab der Version 1.3.

Für die Visualisierung werden je nach der Art der Visualisierung zwei Komponenten benötigt. Bei der Darstellung der Bewegungen mittels VRML ist ein entsprechender Player erforderlich, hierzu wurde Cosmo Player 2.1 verwendet, prinzipiell kann jedoch auch jeder andere Player für VRML-Dateien verwendet werden. Für die zweite Art der Visualisierung ist ein Java3D Package von Sun erforderlich. Es muss zumindest die Version 1.3.1 des Java3D SDK installiert sein.

Zum Verbindungsaufbau wird die Software des Bluetooth Dongle verwendet. Hierbei wird ein Treiber von Widcomm verwendet. Nach der Installation muss die Verknüpfung mit einem COM Port erfolgen. Aufgrund einer Einschränkung in der Software von InterSense zum Verbinden mit dem Sensor muss ein COM Port zwischen eins und acht gewählt werden, da sonst der Sensor nicht gefunden wird und demzufolge auch keine Verbindung zum Sensor hergestellt werden kann. Diese Einschränkung betrifft jedoch nur die Realisierung mit diesem Sensor und der damit verbundenen Software.

Bei der verwendeten Software des Sensors handelt es sich um eine DLL. Hierbei wurden die Versionen 3.44 und 3.58 verwendet, wobei für den Tablet PC und der damit verwendeten Windows Version, zwingend zumindest die Version 3.58 der DLL erforderlich ist.

5.10 Eigenschaften verschiedener geometrischer Formen

Aufgrund der Form ergibt sich bei fast jedem Artefakt des täglichen Lebens eine bevorzugte Art der Manipulation, diese favorisierte Vorgehensweise soll auch bei der Entwicklung von Programmen und TUIs genutzt werden um die Akzeptanz der Benutzer zu steigern. Um die Möglichkeiten die sich dadurch ergeben zu testen wurde die Sensorik in insgesamt sechs verschiedene Formen integriert. Diese Formen stehen somit natürlich auch für die Anwendungen zur Verfügung um eine möglichst optimale Form der Manipulation in Abhängigkeit der Funktionen der Anwendung zu erreichen. Um die Elektronik fest mit dem Gehäuse zu verbinden wurde eine spezielle Halterung für jede der Formen, bis auf den Quader da es hier aufgrund des eingeschränkten Platzes nicht notwendig war, entwickelt die den Sensor so befestigt, dass keine Störungen durch Verrutschen des Sensors oder durch leichte Prellungen entstehen können. Weiters sind auch Montagemöglichkeiten für das Bluetoothmodul und die Akkupacks gegeben, wobei die Akkus einfach gewechselt werden können da sie aus der eigens konstruierten Halterung genommen werden können sobald die Form geöffnet wurde. Zusätzlich wurde eine Möglichkeit zur Befestigung eines Schalters vorgesehen um die Schaltung ein- bzw. auszuschalten, ohne dass die jeweilige Form geöffnet werden muss.

Die sechs Formen und die daraus resultierenden Möglichkeiten für die Interaktion werden im weiteren Verlauf näher beschrieben.

5.10.1 Kegel

Der Kegel hat eine Höhe von 13cm und an der Grundfläche einen Durchmesser von ebenfalls 13cm. Aufgrund des Verhältnisses Durchmesser/Höhe ist der Kegel relativ stumpf, dies hat zur Folge, dass die Spitze des Kegels nicht sehr gut in der Hand liegt, dies fällt vor allem dann auf, wenn die Form in der Hand gehalten wird und keine Ablagefläche (z.B. ein Tisch) genützt wird.

Sowohl im stehenden als auch im liegenden Zustand können die Drehbewegungen genutzt werden, wobei im liegenden Zustand (Spitze als Fixpunkt) eine relativ große Fläche zur Verfügung stehen muss und bei ungünstiger Gewichtsverteilung und glatter Oberfläche die Spitze verrutschen kann. Im stehenden Zustand kann die Drehbewegung sehr gut dosiert werden und wirkt natürlich, jedoch kann durch bloßes Hinsehen nicht erkannt werden wie weit der Kegel bisher gedreht wurde. Aufgrund der runden Grundfläche ergeben sich jedoch beim kippen Probleme, dies ist nur mit einem verhältnismäßig großen Kraftaufwand möglich und führt zu Unsicherheiten in der Manipulation.

Nicht zuletzt aufgrund der Größe ist der Kegel eher dazu geeignet mit einer Unterlage und nicht direkt in der Hand verwendet zu werden.



Abbildung 32: Kegel mit Elektronik

5.10.2 Pyramide

Die Pyramide verfügt über eine Grundfläche mit der Seitenlänge von 11cm und ist insgesamt 19cm hoch. Eine Drehung im stehenden Zustand ist nicht so einfach möglich wie beim Kegel, dafür können jedoch Unterschiede durch das Drehen um die z -Achse (im stehenden Zustand der Pyramide) visuell eingeschränkt wahrgenommen werden. Das Kippen um eine der vier Kanten der Grundfläche ist sehr leicht möglich und ähnelt der Bedienung eines Joysticks.

Eine weitere Möglichkeit ist die Spitze im liegenden Zustand zu nutzen, da hierdurch sehr einfach visuell unterschiedliche Zustände dargestellt werden können, auch hier kann die Pyramide einfach um die Kanten die auf einer Unterlage liegen gekippt werden. Die Spitze könnte beispielsweise den Stundenzeiger einer Uhr oder eines anderen Gerätes bei dem die Ausrichtung von Bedeutung ist repräsentieren.

In die Hand genommen kann der Spitz der Pyramide auch als Zeigegerät oder ähnliches verwendet werden, auch wenn hierzu die Pyramide im Idealfall etwas kleiner sein sollte um leichter in der Hand Platz zu finden.



Abbildung 33: Pyramide mit Elektronik

5.10.3 Quader

Hinter dem Quader steckt eine einfache Zigarettenschachtel, diese wurde gewählt, da sie eine sehr bekannte Form aus dem täglichen Leben ist und somit auch Nichtrauchern ein Verständnis über die Größe und den Möglichkeiten bei der Manipulation gibt. Es werden zwei unterschiedlich lange Schachteln verwendet, eine mit rund 80mm und eine mit rund 100mm. Die Höhe beträgt jeweils 22mm und die Breite 55mm. Beim Quader handelt es sich um die vom Volumen her kleinste aller untersuchten Formen, dies ist nicht zuletzt dadurch begründet das die realisierte Sensorik den Platz in diesen Quadern sehr gut ausfüllt und somit kaum in kleinere Artefakte integriert werden kann.

Der Vorteil des Quaders liegt in den drei unterschiedlichen Seitenlängen, dadurch können viele unterschiedliche Zustände auch visuell unterschieden werden. Diese Zustände beschränken sich nicht nur darauf auf welcher Seite der Quader liegt, sondern es können auch noch unterschiedliche Ausrichtungen wahrgenommen werden. Das Drehen und Kippen ist in solch einer Situation sehr einfach möglich und erlaubt es somit die Zustände des Quaders einfach zu wechseln.

Die Schachtel liegt sehr gut in der Hand, dadurch können mit der Schachtel alle Bewegungen durchgeführt werden, die mit der Hand durchgeführt werden können, zusätzlich ist es aufgrund der geringen Größe auch möglich den Quader in der Hand zu drehen, ohne dass eine Ablagefläche oder die zweite Hand dazu benötigt wird.

5.10.4 Würfel

Der Würfel, mit einer Seitenlänge von 84mm, nimmt, da alle Kanten gleich lange sind, eine Sonderstellung ein. Wenn die Oberfläche einheitlich aussieht und die einzelnen Seiten nicht gekennzeichnet sind ist aufgrund der Form keine Unterscheidung möglich auf welcher Seite der Würfel gerade liegt, diese Eigenschaft eignet sich etwa für die Navigation in dreidimensionalen Räumen sehr gut. Generell sind hier Steuerungsaufgaben gut zu lösen bei denen der Zustand keine sehr große Rolle spielt, so kann das Kippen um eine Seitenkante bedeuten, dass in den nächsten Zustand gewechselt wird (könnte dem Weiterschalten in einer Anwendung entsprechen), der Zustand kann jedoch nicht durch einen Blick auf den Würfel erkannt werden (wurde schon zwei oder drei Mal weiter geschaltet).

Der Würfel lässt sich aufgrund seiner Form sehr gut auf einer Unterlage verwenden, hier spielt speziell das Kippen um eine der zwölf Kanten eine große Rolle. Trotz der Kanten sind es die Menschen durchaus gewohnt solche Formen auch in den Händen zu halten, obwohl beim Greifen und Tasten meist Rundungen bevorzugt werden.



Abbildung 34: Würfel mit Elektronik

5.10.5 Zylinder (groß)

Mit einer Höhe von 15cm und einem Durchmesser von 11cm kann der Zylinder für diese Anwendungen als verhältnismäßig groß eingestuft werden, dies wird bei der Manipulation negativ empfunden. Aufgrund der Greifmöglichkeiten der Hand wird speziell der große Durchmesser als störend bewertet.

Auf dem Tisch stehend ist eine Drehbewegung am einfachsten durchzuführen, kippen hingegen ist nur sehr eingeschränkt möglich. Im liegenden Zustand sticht das Rollen (links - rechts bzw. vor - zurück) als hauptsächliche Interaktionsmöglichkeit hervor. Aufgrund der nicht optimalen Gewichtsverteilung der Hardware im Zylinder bleibt dieser nicht immer in der Position liegen, sondern dreht sich wieder zurück, dies könnte jedoch für bestimmte Anwendungen durchaus gewollt sein.

Eine Manipulation des Zylinders mit einer Hand ohne Unterlage ist aufgrund der Größe kaum mehr möglich.



Abbildung 35: Großer Zylinder mit Elektronik

5.10.6 Zylinder (klein)

Beim kleinen Zylinder handelt es sich eigentlich um ein Barglas mit einer Höhe von 15cm. In Wirklichkeit handelt es sich um einen Kegelstumpf da der Durchmesser 55mm bzw. 51mm beträgt, jedoch wird es aufgrund des kleinen Unterschiedes und der damit verbundenen Ähnlichkeit mit einem Zylinder hier als Zylinder geführt.

Auf einer Unterlage stehend kann eine Drehung mit den Fingern durchgeführt werden, ohne das Handgelenk zu bewegen, dies ist eine Geste die man in Gaststätten immer wieder beobachten kann. Aufgrund der Form kann natürlich nicht erkannt werden, wie weit das Glas bereits gedreht wurde. Das Kippen ist im stehenden Zustand, ist im Gegensatz zum großen Zylinder, sehr wohl möglich.

Für das Rollen im liegenden Zustand gilt im Wesentlichen das Gleiche wie beim großen Zylinder. In der Hand liegt das Glas beinahe wie ein kurzer Stab, es sind also Zeigebewegungen durchaus natürlich, aufgrund der geringen Größe sind die Möglichkeiten der Hand, wie bereits beim Quader, nicht eingeschränkt.



Abbildung 36: Kleiner Zylinder mit Elektronik

5.10.7 Mögliche Interaktion mit den Artefakten

Aufgrund der zuvor beschriebenen Formen lassen sich gewisse Gemeinsamkeiten in der Manipulation in Abhängigkeit der Eigenschaften der Formen finden. Diese Betrachtung ist unabhängig von den Bewegungen die mit einem Artefakt in der Hand durch die Hand durchgeführt werden können, hierzu muss der Körper nur klein genug sein. Eine Unterscheidungs- und somit Gruppierungsmöglichkeit bietet das Vorhandensein von Rundungen. So lassen sich die runden Formen (beide Zylinder, Kegel) besonders gut drehen, für das Kippen sind jedoch gerade Kanten besser geeignet.

Ein weiteres Merkmal liegt in der Unterscheidbarkeit der einzelnen Seiten. Hier ist es jedoch möglich, dass dies für eine Form nur teilweise gilt. Ein Quader hat beispielsweise (bis zu) drei unterschiedlich große Flächen, insgesamt sind es jedoch sechs Flächen. Es können anhand der Seite die auf dem Tisch liegt drei Zustände unterschieden werden. Diese Zahl (Anzahl der unterschiedlich großen Flächen) könnte als Maßzahl verwendet werden. Beim Würfel wäre diese Zahl eins, beim Zylinder und der Pyramide zwei. Desto unregelmäßiger ein Körper ist, desto höher ist diese Maßzahl. Durch eine, z.B. farbliche, Kennzeichnung der Oberflächen kann diese Zahl bei jedem Körper bis auf die Anzahl der Flächen erhöht werden, wobei dies nicht die Absicht von TUIs ist da hier Gegenstände aus dem Alltag verwendet werden sollen und nicht speziell konzipierte Eingabegeräte, es sollen rein die Möglichkeiten beschrieben werden.

Neben der Form spielt auch die Größe eine wichtige Rolle. Die beiden kleinsten Körper lassen sich am Einfachsten verändern, da es sich hierbei genau um die Größe handelt die ohne Probleme in der Hand Platz findet, somit ist die Benützung ohne zusätzliche Unterlage möglich. Neben diesem Kriterium, das durch die Größe beeinflusst wird, kann auch noch unterschieden werden ob eine Bewegung mit einer Hand durchgeführt wird, oder ob dazu eine Zweite benötigt wird. Diese Betrachtung kann noch um das Vorhandensein einer Unterlage erweitert werden. So ist etwa das Drehen des großen Zylinders am Tisch mit einer Hand ohne Probleme möglich, ohne Tisch sind für das Drehen jedoch beide Hände erforderlich.

Aus den bisherigen Eigenschaften geht auch eine andere Möglichkeit zur Auswahl des Körpers in Abhängigkeit der Anwendung hervor. Abhängig davon, ob es sich bei der Art der erforderlichen Eingabe um eine diskrete oder kontinuierliche Steuerung handelt, können unterschiedliche Formen verwendet werden. Als Eingabemöglichkeit für kontinuierliche Steuerungen haben sich runde Formen bewährt, vor allem das Drehen der Zylinder und des Kegels, aber auch das Rollen der Zylinder eignet sich besonders gut. Für diskrete Werte sind

die Körper mit geraden Kanten besser geeignet, da dieser Zustand auch vom Körper eingenommen wird und eindeutig von anderen diskreten Zuständen unterscheidbar ist. Die Kombination von zumindest jeweils einem Objekt von beiden Arten eröffnet somit eine sehr umfangreiche und vielfältige Möglichkeit zur Interaktion da hierbei die speziellen Eigenschaften der jeweiligen Formen ausgenutzt werden können.

5.10.8 Möglichkeiten der Steuerung im Vergleich zu GUI

Hierbei sollen die umfangreichen Einsatzmöglichkeiten dieser Hardware gezeigt werden, die sich durch die große Anzahl von Gesten die erkannt werden können ergeben. Es soll jedoch nicht bedeuten, dass mit TUIs traditionelle GUI-Anwendungen gesteuert werden sollen, dieser Ansatz wurde nur zur einfacheren Darstellung gewählt.

- *Drehen:* Durch Drehbewegungen lassen sich in Abhängigkeit um welche Achse diese Drehung stattfindet die Steuerung von unterschiedlichen GUI Elementen nachahmen. Ein Zylinder kann im liegenden Zustand durch vor- und zurückrollen bzw. hin- und herrollen die Funktionalität eines vertikalen bzw. horizontalen Scrollbalken oder eines Schieberegler einnehmen. Im stehenden Zustand können Drehschalter und -knöpfe nachgeahmt werden.
- *Kippen/umdrehen:* Bei Rechwinkeligen Körpern bedeutet diese Kippbewegung ein drehen um eine Kante, also um 90° , beim Umdrehen wird der Körper zwei Mal um die gleiche Achse gekippt. Abhängig von der Kipprichtung könnten bestimmte diskrete Ereignisse ausgeführt werden. Als Beispiel können hier die nächste/vorhergehende Seite, Seite rauf/runter oder auch linke/rechte Registerkarte relativ zur aktuell ausgewählten angeführt werden.
Je nachdem um welche Achse das Umdrehen erfolgt, können wieder drei, bzw. wenn die Drehrichtung betrachtet wird sogar sechs, unterschiedliche Aktionen ausgeführt werden. Bei einer Standardabfrage mit den Auswahlmöglichkeiten "Ja", "Nein" und "Abbrechen" könnte das Umdrehen nach vorne bzw. hinten als "Ja" gewertet werden (analog zum zustimmenden Nicken des Kopfes), nach links oder rechts als "Nein" (wiederum analog zur ablehnenden Kopfbewegung) und um die Z-Achse als die dritte Alternative gewertet werden.
- *Schütteln:* Unter Schütteln wird die ruckartige hin und her Drehung um eine Achse verstanden, es handelt sich also nicht um eine reine Translation, da diese vom Sensor nicht erkannt werden könnte. Hiermit können einfach Abfragen mit zwei Möglichkeiten gesteuert werden (Schütteln um die X- und die Z-Achse). Weiters kann damit eine Auswahl abhängig von der Bewegung selektiert oder deselektiert werden. Knöpfe die einen Wert erhöhen bzw. verringern können ebenfalls so nachgeahmt werden, wobei die Anzahl der Erhöhungsschritte davon abhängig ist wie oft geschüttelt wird.
- *Kreiseln:* Hiermit wird eine schnelle Drehbewegung um die Z-Achse verstanden, also beispielsweise wenn der Zylinder mit der Grundfläche auf dem Tisch steht und gedreht wird. Als Unterscheidung zu den anderen Bewegungen kann etwa als minimaler Wert eine dreiviertel Drehung erforderlich sein. Eine Möglichkeit der Abbildung auf normalerweise grafisch dargestellte Funktionen ist das Wiederholen bzw. Rückgängigmachen einer Aktion (Unterscheidung aufgrund der Drehrichtung) oder auch das neu Laden (bzw. abbrechen des Ladens) einer Seite in einem Browser. Aufgrund der Anzahl der Drehungen und der Drehgeschwindigkeit können weitere Unterscheidungen erfolgen.
- *Komplexere Bewegungen:* Hierunter fallen im Prinzip alle weiteren Bewegungen die erkannt werden können, also beliebige Folgen von Drehungen um die drei Achsen.

Abhängig von der Bewegung kann jede beliebige Funktion damit realisiert werden, wie etwa "Datei öffnen", "Neue Datei", "Datei speichern", "Datei drucken" usw..

- *Platzieren:* Wenn der Körper in eine bestimmte Position gebracht wird, so kann dies ebenfalls eine Aktion hervorrufen, so könnte ein System dadurch in den Stand-by-Betrieb wechseln. Weiters ist hier auch die Steuerung eines Zeigegerätes (z.B. des Mauszeigers) möglich.
- *Aufheben:* Wird das Artefakt nach längerer Zeit zum ersten Mal wieder bewegt so kann durch die geänderten Sensorwerte ein Gerät ein bzw. ausgeschalten werden, die Hintergrundbeleuchtung eines Displays wird eingeschaltet oder ähnliches.

6 Szenarien / Beispielapplikationen

Anhand einiger realisierter Beispiele sollen die Möglichkeiten des hier verfolgten Ansatzes gezeigt werden. Auf diesen Teil wurde ein großes Augenmerk gelegt da dies als Begründung für die Anstrengungen auf diesem Teilgebiet der Bewegungserkennung oder allgemeiner sogar im gesamten Bereich der TUIs gesehen werden kann.

Die Anwendungen zeigen die zentralen Möglichkeiten des Systems welches im letzten Kapitel beschrieben wurde. Hierbei ist auch zu erkennen, dass die gewählte Lösung sehr vielfältig verwendbar ist, dies spiegelt sich in den vom Ansatz her sehr unterschiedlichen Anwendungen und den damit verbundenen Einsatzgebieten wieder.

In den drei Anwendungen wird auch gezeigt wie die Vorteile der unterschiedlichen geometrischen Formen für die Interaktion genutzt werden können bzw. sogar wie sich die Vorteile der verschiedenen Formen in einem einzigen Szenario so koppeln lassen das das daraus resultierende Gesamtsystem davon entsprechend profitiert. Natürlich wird aufgrund der Beispiele auch deutlich ersichtlich wie wesentlich es ist das solch ein System autonom betrieben werden kann. Die Szenarien wurden so gewählt, dass ein möglichst großer Teil der Möglichkeiten des Frameworks dargestellt wird und gleichzeitig auf unterschiedliche Anwendungsgebiete mit den entsprechenden Anforderungen eingegangen wird. Es ist jedoch nicht möglich alle Ansatzgebiete für solche Systeme abzudecken, sondern es können nur Beispiele für einzelne Bereiche herausgegriffen werden.

Trotz all der Bemühungen können diese Anwendungen nicht so gesehen werden, dass sie nun eins zu eins ins tägliche Leben übernommen werden können, sondern sie sollen vielmehr als kleiner Schritt in eine Welt mit TUIs und als Vorschlag für eine weitere Diskussion auf diesem Gebiet gesehen werden.

6.1 Visualisierung

Der Aufwand für das Erstellen der hier gezeigten Visualisierungen ist sehr gering, da diese Funktionalität bereits durch das im Rahmen dieser Arbeit erstellte API abgedeckt wird. Es wird somit die im API enthaltene Funktionalität nur mehr entsprechend konfiguriert und anhand einer aussagekräftigen Bewegung gezeigt. Der Visualisierungsteil besteht aus zwei großen Blöcken, der Visualisierung mittels VRML Datei und mittels Java3D, wobei es nur die zweite Möglichkeit erlaubt die Sensorwerte und somit die Bewegung dynamisch darzustellen und zu betrachten.

6.1.1 VRML

Diese Anwendung soll eine graphische Darstellung der Sensorwerte ermöglichen, wobei die Visualisierung nicht dynamisch erstellt werden muss. Die Bewegungen und die damit verbundenen Änderungen der einzelnen Sensorwerte sollen über einen längeren Zeitraum möglichst verständlich dargestellt werden, hier ist die Darstellung von Häufungen der Sensorwerte von Interesse.

Es ist nicht möglich die VRML Datei zu betrachten während der Sensor bewegt wird, da die Datei erst nach der Aufzeichnung vollständig geschrieben wird und somit kann die Datei im Viewer erst betrachtet werden nachdem die aufzuzeichnenden Bewegungen durchgeführt wurden, es ergeben sich aus diesen Ansatz jedoch andere Einsatzgebiete. Eine mögliche Anwendung besteht in der Auswertung von länger andauernden Aufzeichnungen. Es können die Sensorwerte beispielsweise ein oder zwei Stunden bei beliebigen Tätigkeiten aufgenommen werden. Aufgrund der Häufungen des Linienzuges, dieser stellt die Änderungen

der Sensorwerte dar, an einzelnen Stellen oder auch in bestimmten Bereichen können einfach ohne Berechnungen und Analysen der Sensorwerte einzelne Bewegungen erkannt werden die wiederholt durchgeführt wurden, oder es kann eine bevorzugte Orientierung des Objektes herausgefunden werden. In einem Versuch bei dem der Sensor auf dem Handrücken befestigt war konnten solche Häufungen erkannt werden. In einer Bürosituation bei der hauptsächlich mit der Tastatur geschrieben wurde und nur gelegentlich ein Blatt Papier geholt oder zwischendurch kurz telefoniert wurde konnte diese Häufung beim Machinschreiben eindeutig erkannt werden. Von dieser extrem dichten Ansammlung von Linien zeigten nur wenige Linien weg, hierbei handelte es sich um die anderen Tätigkeiten, (telefonieren usw.) bei denen die Orientierung der Hand stärker geändert wurde. Dieses Wissen könnte zur Gewinnung von Kontextinformation genutzt werden, da es ein ähnlicher Ansatz zu den Beispielen in Kapitel 2.7.5 ist. Auch wenn die Auswertung in so einem Fall nicht visuell erfolgen würde, so könnten zumindest die Tätigkeiten dargestellt werden. Diese Möglichkeit gibt es natürlich auch wenn die Sensorik in einem alltäglichen Objekt integriert ist. Was mit dem Gegenstand gemacht wird und wie es gemacht wird kann zusätzliche Informationen für die Anwendungen oder auch über den Anwender liefern, diese Informationen können wiederum in dieser Form dargestellt werden.

Als Beispiel wurde zum besseren Darstellen der Möglichkeiten bei dieser Visualisierungsart eine relativ einfache Bewegung gewählt. Die Sensorik ist für diesen Versuch in einer Zigarettenschachtel integriert. Die Schachtel liegt mit einer der beiden größeren Flächen auf dem Tisch. Die Bewegung beginnt damit, dass die Schachtel in die rechte Hand genommen wird und ausgehend hiervon um rund 180° um die x-Achse gedreht wird, sodass die Seite die zuvor auf dem Tisch lag nach oben zeigt. Danach wird der Quader so bewegt, dass diese Seite, welche zu Beginn auf dem Tisch lag, über die Schulter zeigt, dies entspricht einer Drehung um die y-Achse. Wenn diese Bewegung flüssig durchgeführt wird kann die soeben erwähnte Unterteilung in zwei Teilbewegungen nicht mehr ganz erkannt werden da der Bewegungsablauf an den zuvor erwähnten Endpunkten verfließ, zusätzlich kommt noch eine leichte Drehung um die dritte Achse hinzu, da die Bewegung beim Ausführen sonst unnatürlich wirkt. Die Bewegung soll also das Aufnehmen eines Gegenstandes und das "über die Schulter werfen" darstellen und ist in Abbildung 37 zu sehen.

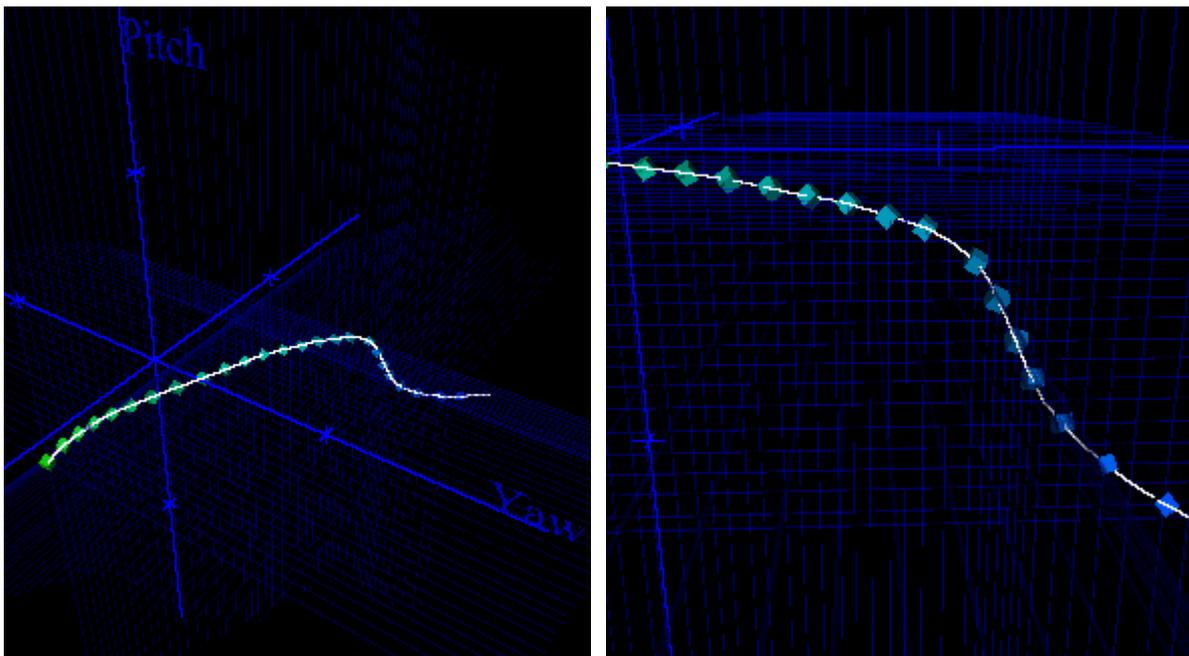


Abbildung 37: Visualisierung einer Bewegung mittels VRML Linienzug. Gesamte Bewegung (links) und Detail dieser Bewegung (rechts)

Im linken Bild handelt es sich um die gesamte Bewegung, rechts ist ein Teil in einer Detailansicht dargestellt.

Die Darstellung ist auf den ersten Blick gewöhnungsbedürftig, da auf den Achsen nicht wie sonst üblich die Position im Raum sondern die Lage des Sensors aufgetragen ist, das bedeutet das jeder der drei Freiheitsgrade des Sensors auf einer Achse des Koordinatensystems aufgetragen ist. Die Länge der Achse in der Visualisierung entspricht einer ganzen Umdrehung des Sensors. Daraus ergibt sich das jeder mögliche Punkt im Diagramm einer bestimmten Lage des Sensors entspricht. Die Bewegung wird durch die weiße Linie dargestellt, wobei die Entfernung der Punkte oder die Farbe der Würfel keine Information über den absoluten zeitlichen Verlauf der Bewegung liefern. Die Entfernung der Punkte zueinander zeigt nur an, wie stark der Sensor gedreht wurde bzw. wie unterschiedlich die Orientierung in den Punkten ist und ist davon abhängig wie oft ein Sensorwert zur Visualisierung hinzugefügt wird. Der Farbverlauf der Würfel liefert nur eine relative Zeitinformation, also welchen Verlauf die Bewegung genommen hat, jedoch kann daraus nicht auf die Geschwindigkeit geschlossen werden. In diesem Beispiel wurde mit einem grünen Würfel begonnen der die Farbe allmählich auf blau ändert, in Abbildung 38 ist eine detaillierte Darstellung des Beginns des Farbverlaufes zu sehen. Zusätzlich kann bei dieser Detailansicht erkannt werden, dass sich neben der Position auch die Lage der Würfel ändert. Die Orientierung der Würfel entspricht der Orientierung des Sensors, diese Information wird somit doppelt dargestellt, da auch die Position des Würfels von der Lage des Sensors abhängig ist. Im konkreten Fall hilft diese Darstellung nur mäßig, da bei einem einfärbigen Würfel die Unterschiede kaum erkannt werden können und eine Unterscheidung der einzelnen Seiten des Würfels nicht möglich ist. Diese Darstellung kann jedoch auch genutzt werden wenn beispielsweise eine Pyramide anstelle der Würfel gezeigt wird und auch der Sensor in einer Pyramide eingebettet ist, hierbei ist leicht ersichtlich das die Darstellung eine größere Aussagekraft hat, da die Richtung in die die Spitze zeigt gut erkannt werden kann und diese Richtung mit der Richtung der Pyramide übereinstimmt die manipuliert wird. Die Anzahl der Würfel und die Abstände können vom Benutzer festgelegt werden und werden nicht zwangsweise für den ganzen Verlauf der Bewegung dargestellt, da dies speziell bei längeren Aufzeichnungsintervallen dazu führen würde, dass die gesamte Darstellung unübersichtlich wird.

In den Abbildungen sind neben dem Koordinatensystem jeweils auch blaue transparente Linien, die in den Abbildungen jedoch wie strichlierte Linien aussehen, zu sehen. Diese transparenten Koordinatensysteme haben ihren Ursprung jeweils im Mittelpunkt der zuvor erwähnten Würfel. Dadurch kann bei entsprechender Perspektive die Bewegung besser dargestellt werden, im rechten Bild in Abbildung 38 ist dies zu sehen. Im linken Bereich dieses Bildes kann sehr gut eine Wellenform erkannt werden, diese repräsentiert die Änderungen einer Achse des Sensors. Aufgrund der großen Anzahl dieser Linien entsteht der Eindruck einer Fläche, dadurch wird es möglich die Information im 3D-Raum anschaulicher darzustellen. Durch die Transparenz dieser Koordinatensysteme ist der eigentliche Linienzug trotzdem noch sehr gut zu erkennen.

Die Art der Modellierung bietet den zusätzlichen Vorteil das mittels VRML-Player die Perspektive innerhalb dieser Darstellung beliebig geändert werden kann und eine Navigation durch diese virtuelle Welt möglich ist, hierbei wird auch der Verlauf der Bewegung und die zeitliche Veränderung durch den Farbverlauf anschaulicher dargestellt als es mittels Screenshots möglich ist.

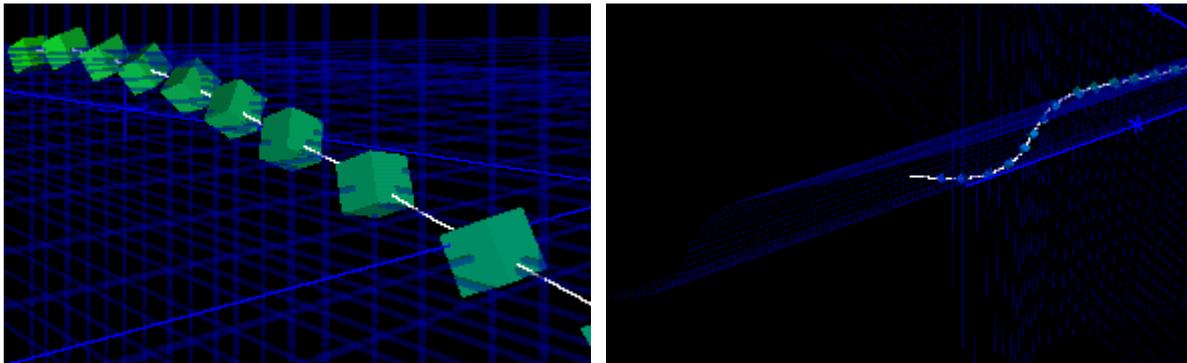


Abbildung 38: Detailansichten der Darstellung einer Bewegung mittels VRML

Die Anforderungen an diese Visualisierung wurden in allen wesentlichen Punkten erfüllt. Durch die transparenten Koordinatensysteme und der abhängig vom Sensorwert gedrehten Würfeln wird es ermöglicht die teilweise widersprüchlichen Anforderungen nach einer möglichst guten Visualisierung der Änderungen der Sensorwerte und der Darstellung der Werte über einen längeren Zeitraum nachzukommen. Dass die Orientierung als Punkt in einem Koordinatensystem dargestellt wird ist zu Beginn, wenn jemand zum ersten Mal einen Kontakt hiermit hat, etwas verwirrend. Dieser Punkt könnte entsprechend verbesserungsfähig sein.

Eine dynamische Betrachtung der Sensorwerte wäre durchaus wünschenswert, wenn auch, aufgrund der Fokussierung auf die Betrachtung von längeren Wertereihen, nicht unbedingt erforderlich. Zusätzlich kann der Verlauf des Sensorwertes besser nachvollzogen werden, da durch den VRML-Player die Perspektive beliebig verändert werden kann.

6.1.2 Java3D

Das Ziel der zweiten Visualisierung ist es dem Benutzer ein Gefühl für die Auswirkungen der Bewegungen zu geben. Hierzu ist es erforderlich dass die Sensorwerte dynamisch dargestellt werden könne, um somit die Effekte der Bewegungen im vollen Umfang darzustellen. Im Gegensatz zur vorhergehenden Visualisierung sind hier längerfristige Darstellungen nicht von Interesse. Dieser Demonstrator ist somit primär zur Darstellung der Funktionalität der Sensorik gedacht.

Als zweite Form der Visualisierung wurde die direkte Darstellung der aktuellen Sensorwerte mittels Java3D API gewählt. Ein wesentlicher Vorteil gegenüber der zuvor erwähnen Möglichkeit ist, dass die Sensorwerte unmittelbar nach der Messung dargestellt werden können, somit werden die Bewegungen sofort sichtbar und auch die Größe der Änderung wird gezeigt. Durch diese Art der Visualisierung kann die Dynamik die in den unterschiedlichen Bewegungen steckt besser dargestellt werden.

Zur besseren Vergleichbarkeit wurde hier wieder die gleiche Bewegung wie bereits bei der Darstellung mittels VRML-Datei gewählt, es handelt sich also um die Bewegung bei der ein Gegenstand aufgehoben und gedreht wird und anschließend Richtung Schulter bewegt wird, als wenn er über die Schulter geworfen wird. Bei der Durchführung der Geste war der Sensor wieder in einem Quader befestigt. Diese Geste wird in Abbildung 39 aus zwei unterschiedlichen Perspektiven gezeigt.

Die Art der Repräsentation der Sensorwerte im Koordinatensystem ist wie bereits bei der Darstellung mithilfe der VRML-Datei. Jede der drei Achsen des Koordinatensystems entspricht einem Freiheitsgrad des Sensors, auf jeder der Achsen wird einer der drei Sensorwerte aufgetragen, somit definiert jeder der Punkte des Koordinatensystems eine eindeutige Lage des Sensors.

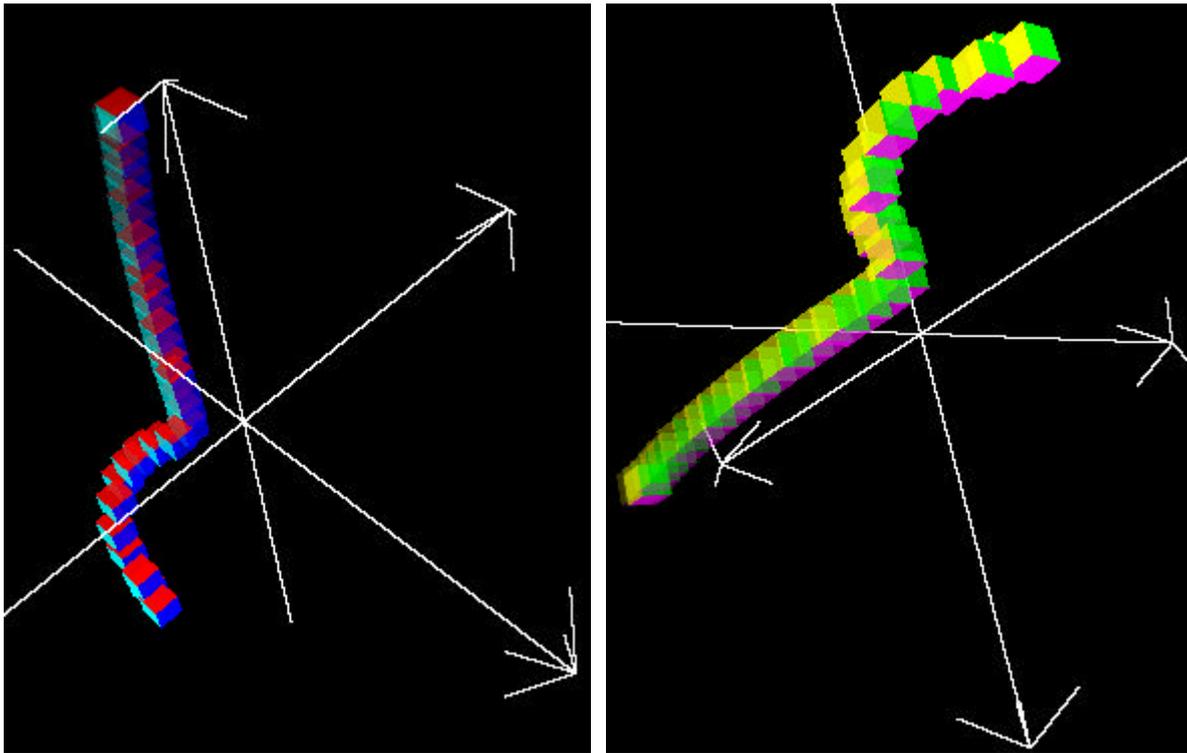


Abbildung 39: Visualisierung einer Bewegung mittels Java3D mit verschiedenen Perspektiven

Die Sensorwerte werden hier nur durch die Würfel dargestellt und nicht zusätzlich durch einen Linienzug oder ähnliches. Hierbei gibt es jedoch eine Vielzahl von unterschiedlichen Konfigurationsmöglichkeiten. So kann die Anzahl der Würfel sehr groß gewählt werden, hierdurch kann bei Bedarf ein zum Linienzug ähnlicher Effekt erzielt werden, zumindest wenn die Seitenlänge des Würfels auch noch angepasst wird.

Das Hauptaugenmerk lag in der Gestaltung der zeitabhängigen Darstellung. Grundsätzlich wurden hier zwei unterschiedliche Varianten realisiert, in der Ersten besteht die Möglichkeit die Transparenz der Würfel Zeit gesteuert zu verändern, d.h. sobald ein Sensorwert eintritt der auch als Würfel in der Visualisierung dargestellt werden soll, wird an der dem Sensorwert entsprechenden Position ein Würfel ohne Transparenz gezeichnet. Die Transparenz wird nun nach einen konfigurierbaren Intervall erhöht bis der Würfel komplett unsichtbar ist, somit kann auch die Geschwindigkeit der Bewegung dargestellt werden, da Sensorwerte die früher aufgetreten wurden eine höhere Transparenz vorweisen als aktuellere Werte. In der Darstellung wirkt dies wie wenn der aktuelle Wert immer einen Schweif nach sich zieht, wobei die Länge dieses Schweifes von der Geschwindigkeit der Bewegung, also wie schnell sich die Sensorwerte ändern, und natürlich von der Anzahl der Würfel die dargestellt werden abhängig ist. Wenn sich die Sensorwerte nicht weiter ändern so wird dieser Schweif immer kürzer und löst sich schließlich auf, es bleibt somit nur noch der Würfel der die aktuelle Orientierung des Sensors repräsentiert.

Bei der zweiten Variante wird immer eine konstante Anzahl von Würfeln und somit auch von Sensorwerten angezeigt. Die Darstellung der Zeitinformation erfolgt wiederum durch die zunehmende Transparenz der Würfel, wobei hierbei nur gezeigt werden kann, welche Sensorwerte früher eingetreten sind, es ist somit nur ein relatives und kein absolutes Maß für die Zeit. Im Fall, das der Sensor nicht mehr bewegt wird, bleibt der Schweif der durch die letzten Sensorwerte festgelegt wird trotzdem erhalten und es gibt auch keine Änderungen bei der Transparenz.

Die unterschiedlichen Farben der sechs Seiten des Würfels haben keine bestimmte Bedeutung, sie dienen primär zur besseren Orientierung von welcher Perspektive die Szene

aktuell betrachtet wird. Es ist auch in dieser Visualisierungsform möglich, die Richtung von der das Koordinatensystem betrachtet werden soll frei zu wählen, zusätzlich kann auch der Betrachtungspunkt durch die freie Navigation in der Szene beliebig verändert werden, das alles natürlich während die aktuellen Sensordaten dargestellt werden.

Da die Darstellung hier nur durch Screenshots erfolgen kann, geht die dynamische Komponente verloren, somit kann ein wesentlicher Teil dieser Anwendung nur schwer zur Geltung gebracht werden.

Durch die verwendete Technologie können die Bewegungen sehr gut zeitabhängig dargestellt werden. Es wird auch gezeigt, wie schnell und wie sensitiv die Anwendungen auf eine Drehung des Sensors reagieren können, d.h. es ist mit dem Auge keine Verzögerung zwischen der Drehung und der Änderung der Visualisierung feststellbar und es kann bereits eine sehr geringe Drehung erkannt werden. Auch hier ist die Darstellung der Orientierung in einem Koordinatensystem wieder gewöhnungsbedürftig, da man die Auswirkungen der Bewegungen jedoch sofort sieht, wiegt dies in dieser Anwendung nicht so schwer.

Generell kann gesagt werden, dass die Visualisierung für die gedachte Anwendung gut geeignet ist.

6.2 RobotControl

Das Ziel ist es zu zeigen, dass ein TUI wie es in dieser Arbeit entwickelt wurde aufgrund seiner Vielseitigkeit praktisch in beinahe jeder Umgebung und vor allem in jeder Domäne, also in unterschiedlichen Arbeitsgebieten, eingesetzt werden kann.

Diese Anwendung soll nicht als weiterer Versuch der Visualisierung angesehen werden, es handelt sich hier um ein Beispiel das viel weitgreifendere Möglichkeiten aufzeigen soll. Der Grundgedanke wurde so gesehen, dass es auch möglich ist TUIs in industriellen Anwendungen einzusetzen, wobei die Wahl eines Beispiels natürlich keine Einschränkung auf einen bestimmten Teilbereich dieser Domäne bedeutet.

Da eine Realisierung eines solchen Industrieszenarios auch eines entsprechenden finanziellen und zeitlichen Aufwands für die Errichtung der industriellen Umgebung oder eines vergleichbaren Modells bedarf wurde dies kurzerhand durch eine visuelle Darstellung simuliert.

In diesem Szenario lag das Ziel nicht primär in der Darstellung der Möglichkeiten der Bewegungserkennung sondern es sollte der Funktionsumfang der Hardware gezeigt werden. Die Wahl fiel auf ein klassisches Beispiel aus der Automatisierungsindustrie, und zwar auf einen Roboterarm. Da die Sensorik insgesamt drei Freiheitsgrade bietet war es natürlich nahe liegend diese auch entsprechend zu nutzen. Hierbei lag die Art der Abbildung der Sensorbewegungen auf den Roboterarm auf der Hand. Aufgrund der drei Freiheitsgrade der Sensorik wurde der Roboterarm ebenfalls mit drei Freiheitsgraden ausgestattet, somit war eine einfache Zuordnung der Änderungen der Sensorwerte auf die Bewegung des Roboterarms möglich.

Die Visualisierung des Roboterarms basiert auf einem Beispiel aus dem Tutorial [23]. Bei der Modellierung ging es nicht um eine möglichst realistische und schöne Darstellung eines Roboterarms, sondern es reichte eine schlichte Form mit der Möglichkeiten gezeigt werden können. In Abbildung 40 ist der Roboterarm in zwei unterschiedlichen Stellungen zu sehen die sich direkt aus der Sensorlage ergeben.

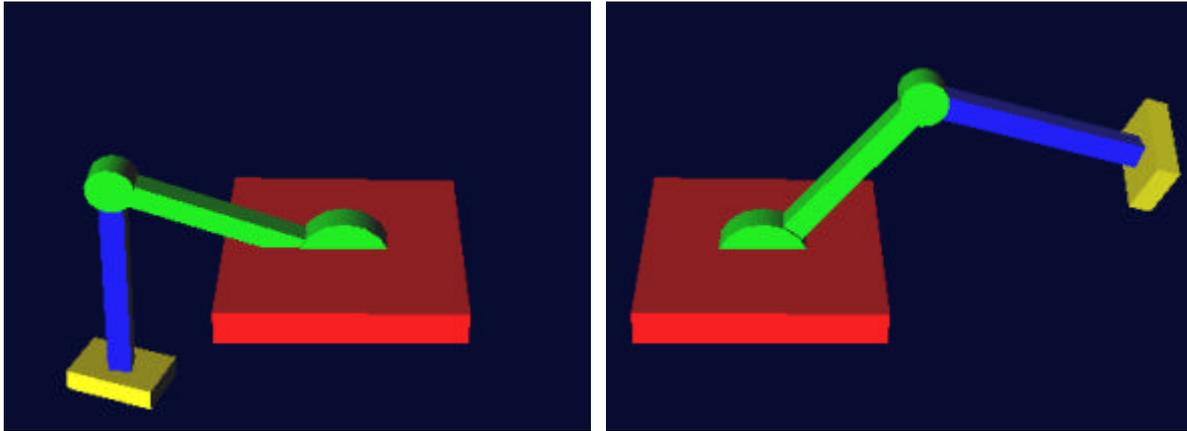


Abbildung 40: Roboterarm in zwei unterschiedlichen Zuständen

Die drei Freiheitsgrade des Roboterarms können leicht erkannt werden. Der Erste befindet sich am Gelenk zwischen der roten Grundfläche und dem grünen Arm, der zweite betrifft das Gelenk zwischen grünem und blauem Arm. Der dritte Freiheitsgrad ermöglicht das Drehen des gelben Quaders. Jeder der Freiheitsgrade des Sensors ermöglicht nun das Drehen einer dieser drei Gelenke, d.h. der Sensorwert wird direkt übernommen und legt die Orientierung des entsprechenden Gelenkes fest. Beim gelben Quader kann diese Abbildung am Besten gesehen werden, da dieser auch in der Simulation um 360° gedreht werden kann, wenn der Sensor um diese 360° gedreht wird. Bei den beiden anderen Gelenken gibt es geringfügige Einschränkungen da etwa der grüne Arm die Grundfläche nicht durchdringen kann, diese Bedingungen wurden jedoch nur für die Simulation eingeführt und sind nicht durch die Sensorik begründet. Die Orientierung des Sensors entspricht somit, bis auf die genannte Einschränkung, immer den Winkeln der Arme des Roboters bezogen auf einen bestimmten Startwert.

Die Elektronik wurde in zwei unterschiedliche Formen integriert, in den Würfel und wiederum in den Quader. Diese beiden Körper riefen bei der Bedienung zwei unterschiedliche Vorgehensweisen hervor. Der Quader wurde bevorzugt mit einer Hand gehalten und die gewünschten Bewegungen des Roboterarms wurden durch die Bewegung der Hand hervorgerufen, erst wenn eine Bewegung durch die Hand nicht mehr möglich war, also etwa eine vollständige Drehung um eine Achse, so wurde der Quader in der Hand gedreht. Die zweite Hand wurde nur dann zu Hilfe genommen, wenn besonders feine Änderungen gewünscht waren oder beim anfänglichen experimentieren wie sich die einzelnen Bewegungen auf den Roboterarm auswirken. Beim Würfel wurde hingegen bevorzugt mit beiden Händen gearbeitet. Dadurch wurden die Bewegungen die der Roboter ausführen sollte primär durch direktes Drehen des Würfels ausgelöst, ohne dass die Hände die gleiche Bewegung mitmachten. Die Lage und Position der Hände änderte sich während der gesamten Manipulation mit dem Würfel im Vergleich zum Quader relativ wenig. Diese unterschiedliche Handhabung liegt neben der Form wohl primär an der Größe da es beim Quader wesentlich einfacher ist die gesamte Form in einer Hand zu halten.

Eine Anwendung die tatsächlich in der Industrie stattfindet und nicht nur ein solches Szenario simuliert wäre noch besser gewesen, dies ist jedoch aufgrund des Kosten- und Arbeitsfaktors nur schwer zu realisieren. Unter dieser Einschränkung stellt es die geforderten Kriterien allerdings sehr gut dar.

Die Simulation darf derzeit nur als theoretisches Beispiel gesehen werden, da eine Realisierung aus mehreren Gründen wie Sicherheitsbedenken, große Bewegungen des Roboterarms bei kleinen Bewegungen oder einer unruhigen Hand usw., noch nicht durchführbar ist.

6.3 MediaplayerControl

Das Ziel ist eine Anwendung, welche die Möglichkeiten der Gestenerkennung aufzeigt und die grundlegenden Eigenschaften eines TUIs besitzt. Es soll gezeigt werden, dass zwischen einer großen Anzahl von verschiedenen Bewegungen unterschieden werden kann, wobei diese Gesten möglichst natürlich und intuitiv für die gewünschte Aktion zu wählen und an die geometrische Form der Artefakte anzupassen sind. Weiters soll dargestellt werden, dass die Gesten auf eine beliebige Aktion abgebildet werden können, also auch z.B. mehrere Gesten auf eine Aktion.

Durch die Integration in einen alltäglichen Gegenstand soll die Rolle als TUI oder auch als persönliches TUI stärker hervorgehoben werden.

In Abbildung 41 ist eine alltägliche Situation zu sehen, was es hiermit auf sich hat, wird im Folgenden erklärt.



Abbildung 41: Situation in einem Kaffeehaus

Ein Ziel dieses Beispiels ist es, zu zeigen wie einfach Tangible User Interfaces in unsere Umwelt integriert werden können und welchen Nutzen der Benutzer davon hat. Bei diesem Beispiel handelt es sich um die zentrale Anwendung die entwickelt wurde, hierbei wird auch der ursprüngliche Ansatz der TUIs, wie etwa die Integration in alltägliche Objekte, wieder verstärkt hervorgehoben. Das entstandene System nutzt natürlich wieder die Hardware und die damit verbundene Bewegungserkennung, jedoch liegt der Fokus auf dem resultierenden Gesamtsystem und nicht, wie teilweise in den Beispielen zuvor, auf der Darstellung der Möglichkeiten der Sensorik oder auch des Frameworks. Der genaue Zusammenhang mit der in der Abbildung gezeigten Szene wird in Kapitel 6.3.2 erläutert.

Als Ausgleich zur zuvor gewählten Industrienahen Anwendung wurde hier ein Beispiel aus dem Privatbereich gewählt, wobei das durchaus auch in den Bereich Dienstleistung (Gastronomie) Auswirkungen haben kann. Die Idee ist es, Unterhaltungselektronik oder auch die Elektronik der Hausautomatisierung mittels TUIs zu steuern. Da derzeit eine Steuerung der meisten Unterhaltungsgeräte (Fernseher, DVD-Player, Stereoanlage, ...) nur mit einer Fernbedienung möglich ist, wäre eine Umsetzung von den erkannten Gesten auf diese Steuerung erforderlich gewesen, zusätzlich gibt es somit Abhängigkeiten zu den verwendeten Geräten. Zur Darstellung kann jedoch der gleiche Effekt erzielt werden wenn ein Software Player verwendet wird, wenn eine Ausgabe über die Lautsprecher bzw. das Anzeigegerät erfolgen soll, so kann dies auch vom Software Player her durchgeführt werden. Dies ist durchaus ein realistischer Ersatz, da es in absehbarer Zeit wahrscheinlich eine Unterhaltungszentrale für das gesamte Haus bzw. die Wohnung gibt von der man die einzelnen Räume mit

Bild und Ton versorgen kann. Die Anwendungen könnten natürlich um Gesten erweitert werden um auch den zweiten angesprochenen Teil, die Hausautomatisierung, abzudecken. Hierbei könnte man beispielsweise die Raumtemperatur regeln oder auch die Jalousien bedienen. Wenn eine einheitliche Schnittstelle zur Kommunikation mit all diesen zuvor genannten Geräten zur Verfügung steht so ist es prinzipiell auch möglich, dass das persönliche bzw. personalisierte TUI mitgenommen werden kann und dann abhängig von der Position ein Geräte in der Umgebung, bei entsprechender Berechtigung, steuert.

Grob umschrieben handelt es sich um folgendes gewähltes Szenario: Mithilfe der TUIs die in unterschiedlichen Formen zur Verfügung stehen soll ein Mediaplayer bedient werden können, dieser Player läuft auf dem Gerät das auch für die Auswertung der Gesten zuständig ist. Bei Bedarf wird die Auswerteeinheit noch mit einer vorhandenen Tonaanlage oder mit einem Display verbunden.

Bei der Wahl des Software Players gab es grundsätzlich zwei Möglichkeiten. Die Erste ist die Eigenentwicklung eines Players mit einem zumindest minimalen Funktionsumfang die durch ein TUI gesteuert werden kann, die Zweite ist die Verwendung eines bereits bestehenden Players bei dem eine Steuerung durch eine andere Software möglich ist. Aufgrund des besseren Gesamtergebnisses (ausgereifterer Player vs. optimale Anpassung des Funktionsumfangs an den tatsächlichen Bedarf) fiel die Wahl auf einen als Freeware verfügbaren Player, der die Steuerung mittels Plugins erlaubt und sowohl Audio als auch Videodateien abspielen kann. Konkret handelt es sich um Winamp von Nullsoft [58]. Durch die unterstützten Medienarten kann hiermit sowohl ein DVD-Player als auch die zuvor erwähnte Stereoanlage, zumindest die Radio beziehungsweise die CD-Player Funktionalität, natürlich nicht der Verstärker, ersetzt werden. Neben einem großen Funktionsumfang, verschiedensten Gestaltungsoptionen und der Möglichkeit den Großteil der gängigen Mediendateien abzuspielen bietet der Player auch noch die Erweiterbarkeit mittels Plugins an, dies wurde genutzt um den Player durch die Resultate der Bewegungserkennung zu steuern. Bei dem verwendeten Plugin handelt es sich um das ebenfalls frei verfügbare „Moys remote control“ Plugin von Gouarin [34]. Diese Wahl erfolgte, da es möglich ist mittels einer Verbindung über Sockets mit dem Plugin zu kommunizieren. Somit gelingt es auch den Player direkt aus einem beliebigen Programm, in diesen Fall dem Anwendungsprogramm das auch die Bewegungserkennung nutzt, zu steuern.

Trotz des bereits vorhandenen Players ist natürlich eine Ansteuerungssoftware erforderlich die für die Kommunikation mit dem Plugin des Players zuständig ist, die die Bewegungserkennung mit den notwendigen Gesten integriert und zusätzlich speichert, in welchem Zustand sich der Körper befindet, dies ist erforderlich, da Gesten in unterschiedlichen Zuständen des Players unterschiedliche Auswirkungen haben können, so ist beispielsweise das Resultat des Drückens der Pause Taste einer konventionellen Fernbedienung normalerweise davon abhängig ob sich das Gerät bereits im Pause Zustand befindet oder ob aktuell ein Titel wiedergegeben wird. Durch die Speicherung des Zustandes in Kombination mit der Bewegungserkennung wird es auch möglich eine physikalische Repräsentation der digitalen Information, wie es bei TUIs gefordert ist, zu erreichen.

Um einen besseren Überblick über die Möglichkeiten und die Vor- und Nachteile der Steuerung eines Mediaplayers mittels TUIs zu erlangen wurden zwei Versionen mit unterschiedlichen geometrischen Formen implementiert. Diese beiden Anwendungen verfügen nicht über einen komplett identischen Funktionsumfang, dies wäre jedoch auch nicht sinnvoll, da man hiermit die Funktionen nur an die Gegebenheiten der Form anpassen würde und nicht umgekehrt. Es wurde also ein großen Augenmerk darauf gelegt, dass die Möglichkeiten der geometrischen Formen möglichst gut genutzt werden. Der genaue Funktionsumfang und die

Abbildung der Bewegungen auf die Funktionen des Players werden in den jeweiligen Unterpunkten behandelt.

Aufgrund der unterschiedlichen Formen kann es außerdem auch zielführend sein unterschiedliche Metaphern bei der Steuerung der Geräte anzuwenden, dies wurde auch beispielsweise bei der Lautstärkenregelung so gehandhabt. Die genauen Gründe, die verwendeten Metaphern und die Auswirkungen hierfür sind in den beiden Unterpunkten für den jeweiligen Fall erklärt.

Speziell bei der zweiten Realisierungsart der Steuerung (Kapitel 6.3.2) wurde neben der einfachen und intuitiven Interaktion auch auf eine harmonische Integration in die Umgebung großer Wert gelegt, dies spiegelt sich auch in der Wahl der Formen wieder.

6.3.1 Steuerung mittels Pyramide

In dieser Version erfolgt die Steuerung mithilfe der Pyramide, welche bereits in Kapitel 5.10.2 beschrieben wurde. Einer der entscheidenden Gründe bei der Wahl war die visuelle Unterscheidungsmöglichkeit von unterschiedlichen Zuständen, speziell zwischen einer stehenden und einer liegenden Pyramide.

Insgesamt werden in diesen Ansatz folgende Funktionen des Players unterstützt:

- Stopp/Play
- Lautstärkenregelung
- Vorhergehender bzw. nächster Titel
- Schneller Vor- und Rücklauf
- Direkte Titeleingabe mithilfe der Ziffern 0-9

Es werden zwei visuell sehr unterschiedliche Zustände unterschieden, die stehende und die liegende Pyramide. Abhängig vom Zustand sind auch die möglichen Funktionen die von diesem Zustand ausgehend ausgeführt werden können. In Abbildung 42 ist die liegende Pyramide mit den angedeuteten Bewegungen zu sehen.

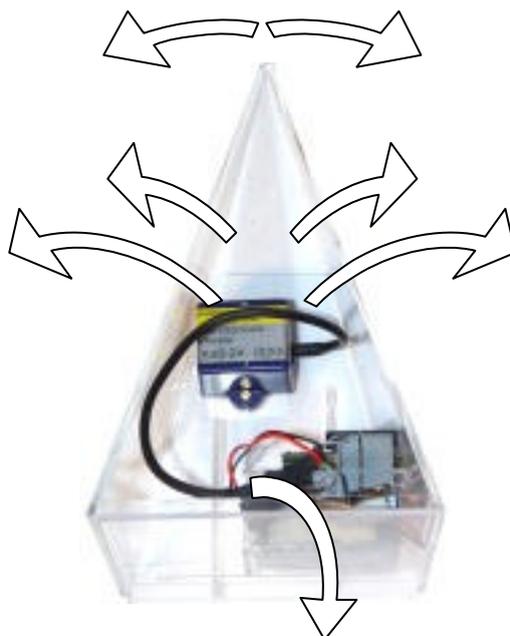


Abbildung 42: Liegende Pyramide mit Bewegungen

Prinzipiell bedeutet der Zustand in dem die Pyramide liegt „Play“, die Bedeutung für den stehenden Zustand ist hingegen „Stop“. Diese Unterscheidung ermöglicht es mit einem Blick auf die Pyramide eine Aussage über den Zustand des Players zu treffen, hierbei wird die digitale Information unaufdringlich durch die Lage der Pyramide repräsentiert. Durch das Aufstellen bzw. Umlegen kann zwischen den Zuständen „Stopp“ und „Play“ umgeschaltet werden, diese Bedeutung ist auch mit dem Pfeil der zum unteren Rand der Abbildung 42 zeigt gemeint.

Durch das Drehen der Spitze nach links bzw. rechts kann leiser bzw. lauter gedreht werden. Die Lautstärke wird verändert sobald ein gewisser Winkel überschritten wird, wenn dieser Winkel wieder unterschritten wird bleibt die Lautstärke wieder konstant. Die Idee dahinter ist das ein- und ausschalten der Lautstärken Änderung als Metapher dient die Lautstärkenregelung bei einer Fernbedienung, hierbei wird die Lautstärke auch so lange geändert bis man den Knopf wieder loslässt. Gewisse Unterschiede gibt es natürlich da durch das loslassen der Pyramide die Lautstärkenänderung nicht aufhört, sondern erst wenn der Spitz wieder in der Ausgangslage ist, wie es auch beim Knopf der Fernbedienung ist, jedoch wird hier der Taster beim Auslassen automatisch in die Ausgangsposition gebracht.

Die beiden weiteren Funktionsgruppen im liegenden Zustand bedienen sich der Kanten der Pyramide. Konkreter geht es um die linke und rechte Kante welche die Auflagefläche berühren und von der Grundfläche zur Spitze führen (Angaben beziehen sich jeweils auf Abbildung 42). Wird die Pyramide leicht um eine der beiden Kanten gekippt, so entspricht das einem schnellen Rücklauf bei der linken Kante und einem schnellen Vorlauf bei der rechten Kante. Die Pyramide wird also in die Richtung bewegt wie es der Anordnung der Tasten bei fast allen Anwendungen mit einer ähnlichen Funktionalität der Fall ist. Dieser schnelle Suchlauf kann, äquivalent zur Lautstärkenregelung, wieder beendet werden, indem die ursprüngliche Lage eingenommen wird. Wird die Pyramide im Suchlauf noch weiter um diese Achse gekippt, so wird, abhängig ob es die linke oder rechte Kante betrifft, gleich zum nächsten Titel zurück- bzw. weitergeschaltet. Wenn mehrere Titel übersprungen werden sollen, muss die Pyramide mehrmals gekippt werden und dazwischen wieder den Ausgangszustand einnehmen, eine Alternative hierzu, die direkte Titelwahl, gibt es im stehenden Zustand der Pyramide. Bei den einzelnen Bereichen, wie etwa beim Ausgangszustand, gibt es natürlich immer entsprechende delta-Bereiche in dem sich der Sensor befinden muss, damit ein Zustand eintritt, um die Handhabung für den Benutzer zu erleichtern bzw. überhaupt erst zu ermöglichen.

Durch die genannten Gesten sind im Play-Zustand, also wenn die Pyramide liegt, alle erforderlichen Funktionen für eine grundlegende Steuerung des Mediaplayers ausführbar. Aufgrund der Wahl der Bewegungen ist es jedoch erforderlich, dass eine Auflagefläche zur Verfügung steht, dafür werden auch gewisse Informationen über den Zustand des Players durch die Lage der Pyramide ersichtlich. Bei einer anderen Wahl der Gesten, mit den entsprechenden Abbildungen auf die Funktionen des Players, bei der keine Unterlage erforderlich ist, wäre die Repräsentation der digitalen Information schwieriger zu erreichen und die Handhabung würde davon auch nicht unbedingt profitieren, speziell wenn als physikalisches Objekt diese Pyramide gewählt wird.

Beim zweiten großen Hauptzustand handelt es sich um den Stopp Zustand, hierbei steht die Pyramide auf der Grundfläche, dieser Zustand ist somit auch visuell eindeutig vom Play Zustand zu unterscheiden. In Abbildung 43 ist die stehende Pyramide mit den angedeuteten Bewegungen zu sehen.

Durch das Umlegen der Pyramide in die Richtung die im Bild mit der Ziffer zwei gekennzeichnet ist kann in den Play Zustand gewechselt werden. Einige der Funktionen wie die Lautstärkenregelung und das weiterschalten der Titel, können in unterschiedlichen

Zuständen (Stopp und Play) ausgeführt werden, es wurden die dazu notwendigen Gesten jedoch an den Zustand angepasst, d.h. es sind abhängig vom Zustand nicht immer die gleichen Gesten erforderlich um eine bestimmte Funktion des Players auszuwählen. Auch im Stopp-Zustand sind wieder alle grundlegenden für die Steuerung des Mediaplayers in diesen Zustand erforderlichen Funktionen vorhanden, so wäre es nicht sehr komfortable wenn man die vom letzten Betrieb eingestellte Lautstärke erst verringern kann nachdem bereits ein Titel abgespielt wird.

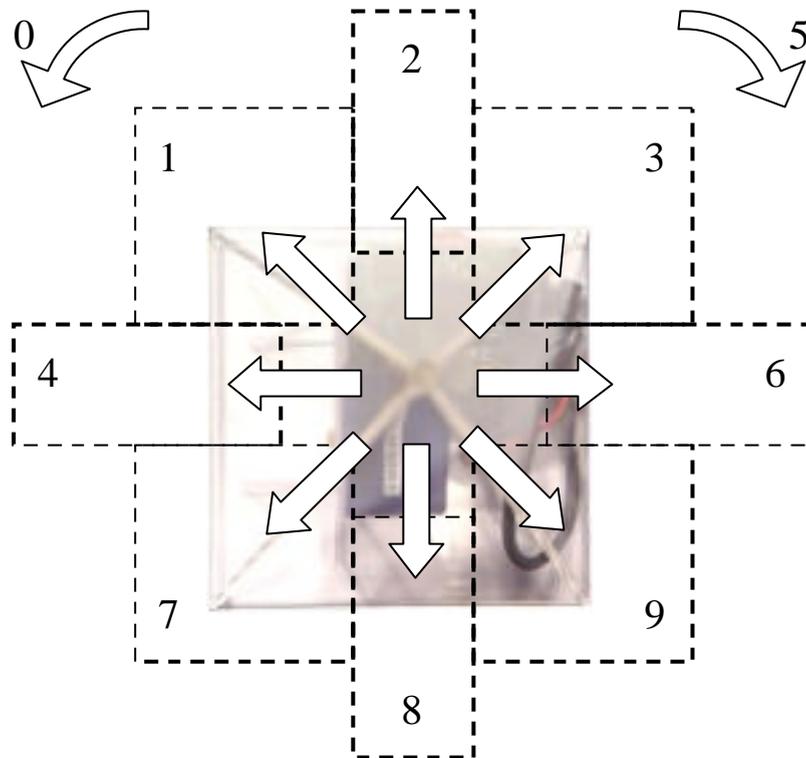


Abbildung 43: Stehende Pyramide mit Bewegungen

Durch leichtes Kippen in die Richtungen vier bzw. sechs wird der vorhergehende bzw. nächste Titel ausgewählt. Die Lautstärkenregelung erfolgt durch drehen nach links (leiser) beziehungsweise nach rechts (lauter), wobei auch hier wieder die Metapher aus dem liegenden Zustand angewandt wird, die Lautstärke verändert sich also solange die Pyramide gedreht ist und ist unabhängig vom Drehwinkel, zumindest wenn ein bestimmter Wert überschritten wurde damit die Geste erkannt werden kann.

In Abbildung 43 sind die Ziffern null bis neun zu sehen, diese betreffen die direkte Eingabe eines Titels. Um diese Funktion nutzen zu können muss zuerst einmal die Pyramide nach hinten, also in die Richtung der Ziffer acht bewegt werden, hiermit wird ausgewählt das nun die Zifferneingabe aktiv ist. Nun kann die Pyramide leicht in die gewünschte Richtung gekippt werden, wenn mehrere Ziffern eingegeben werden sollen, muss zuerst wieder der Ausgangszustand eingenommen werden. Da bei den Ziffern zwei, vier, sechs und acht ein Kippen um eine Kante erfolgt und dies wesentlich einfacher und genauer möglich ist als das Kippen um eine Ecke, konnte hier der Bereich für diese Ziffern, dieser ist durch die strichlierten Felder angedeutet, etwas kleiner gewählt werden als für die Ziffern ein, drei, sieben und neun. Die Größe dieser Flächen soll nur das Prinzip zeigen und muss nicht mit der aktuellen Konfigurierung übereinstimmen. Die Anordnung der Ziffern wurde aufgrund der großen Verbreitung von Telefonen analog dazu gewählt, alternativ wäre auch die Anordnung wie sie am Ziffernblock der Computertastatur ist möglich. Die beiden Ziffern die aufgrund

dieser Struktur nicht ausgewählt werden können, können durch Bewegungen die von diesen Konzept etwas abweichen eingegeben werden, indem die Pyramide nach links (Ziffer null) bzw. nach rechts (Ziffer fünf) gedreht wird, diese Bewegung ist mit den Bewegungen beim Ändern der Lautstärke identisch. Durch die Eingabe von mehreren Ziffern nacheinander kann jeder beliebige Titel aus einer Liste ausgewählt werden. Zur Erleichterung und Erhöhung der Geschwindigkeit bei dieser Eingabeform wird beim Erkennen einer Ziffer ein kurzer Ton als Bestätigung abgespielt.

Bei der Eingabe der Ziffern für die Titelauswahl kann man erkennen welche vielfältigen Möglichkeiten es bei diesem Ansatz der Gestenerkennung es gibt. Ausgehend von einem Startzustand können hierbei mit einfachsten Bewegungen zumindest zehn unterschiedliche Gesten erkannt werden.

Es wurden für diese Anwendung nur sehr einfache Gesten verwendet, somit ist kein Erlernen oder eine spezielle Anpassung der Bewegungen an den Benutzer erforderlich, wenn davon ausgegangen wird, dass keine Ablagefläche zur Verfügung steht würde sich die Komplexität der Gesten auch entsprechend ändern um trotzdem noch alle Funktionen bedienen zu können. Durch diese einfachen Gesten ist es für den Benutzer auch leichter, da es sich hierbei um Gesten handelt die sonst mit dieser geometrischen Form auch durchgeführt werden und es parallelen zur bisher üblichen Steuerung - etwa die Bedeutung von links und rechts bei der Titelwahl - gibt.

Generell bietet dieser Ansatz nach einer sehr kurzen Eingewöhnungszeit einige Vorteile gegenüber den bisher üblichen Steuerungen von solchen Playern. Alleine die Möglichkeit durch das Drehen der Pyramide normalerweise in dieser Form nicht greifbare Informationen manipulieren zu können führt zu einer anderen Art der Interaktion. Die Pyramide benötigt außerdem auch nur dann die Aufmerksamkeit des Benutzers wenn der Zustand des Players geändert wird, sonst steht sie unaufdringlich herum und stellt trotzdem noch bestimmte Informationen des Players dar.

Da hier keine Integration in einen Alltagsgegenstand erfolgte, ist die Pyramide jedoch nur eine Alternative zu andern Möglichkeiten zur Steuerung und ist noch nicht so in die Umwelt eingebunden wie das gewünscht wird. Dies wird jedoch im folgenden Ansatz versucht zu verbessern.

6.3.2 Steuerung durch mehrere TUIs

Die Szene, die Eingangs in diesem Kapitel (Abbildung 41) zu sehen war, dient als Motivation für die zweite realisierte Version des Mediaplayers, da es sich hierbei um eine realistische und oft auftretende Alltagssituation handelt. Ein Ziel von TUIs ist immer diese zusätzlichen Funktionen möglichst natürlich in die Umgebung einzubetten, dies war hier auch ein wichtiges Designkriterium. Die beiden wesentlichen Artefakte aus zuvor erwähntem Bild sind in Abbildung 44 noch einmal hervorgehoben und bereits mit den skizzierten möglichen Gesten versehen.

Bei den beiden Artefakten handelt es sich um die bereits bekannte Zigarettenschachtel und zusätzlich noch um eine Red Bull Dose, es sind somit Gegenstände die tatsächlich aus dem täglichen Leben genommen und mit zusätzlicher Funktionalität erweitert wurden. Der Benutzer ist mit den Formen und den damit verbundenen Manipulationsmöglichkeiten vertraut, die Aufgabe ist es somit diese unterbewusst verwendeten Bewegungen zu finden und mit den damit beabsichtigten Aktionen zu verknüpfen.

Die Dose hat mit den in Abschnitt 5.10.6 vorgestellten kleinen Zylinder gewisse Ähnlichkeiten, jedoch ist sie noch eine Spur kleiner, konkret hat sie eine Höhe von 134mm und einen Durchmesser von 53mm. Aufgrund von technischen Anforderungen musste eine

Änderung an der Dose durchgeführt werden, diese betrifft jedoch nur den Boden. Das Metall der Dose wirkt auf das Bluetoothmodul als sehr gute Abschirmung, dies wird in diesen Fall natürlich nicht gewünscht, so beträgt die Reichweite anstelle der 85-90m unter optimalen Bedingungen beim Einbau des Senders in die Dose nur noch rund 1,5m. Aus diesem Grund wurde der Boden der Dose zum Teil durch einen Boden aus Acrylglas ersetzt, dadurch werden mit der gleichen Hardware wieder Reichweiten von über 70m erreicht, was für die Anwendung mehr als ausreichend ist.



Abbildung 44: Schachtel und Dose mit Bewegungen

Im Gegensatz zu allen bisherigen Beispielen erfolgte hier eine Kopplung von mehreren TUIs, hierauf basieren einige der Vorteile. Einer der Vorteile ist, dass die Eigenschaften der jeweiligen Formen, es werden zwei sehr unterschiedliche Formen verwendet, ausgenutzt werden können und sich somit ein breiteres Spektrum an einfach durchführbaren und erkennbaren Gesten ergibt.

Da der Funktionsumfang an die Möglichkeiten der beiden Formen angepasst wurde, gibt es leichte Unterschiede zur Realisierung mit der Pyramide, so gibt es hier keine Eingabe von Ziffern. Folgende Funktionen wurden realisiert:

- Stopp/Play/Pause
- Lautstärkenregelung
- Vorhergehender bzw. nächster Titel
- Schneller Vor- und Rücklauf
- Auswahl von Playlisten

Bei den Gesten wurde neben dem Ausnutzen der Vorteile der beiden Formen auch versucht, eine klare logische Strukturierung der Aufgabengebiete zwischen den beiden Formen zu realisieren. Es erfolgte eine Aufteilung zwischen der Auswahl der Funktion und dem Verändern der damit verbundenen Werte.

Bei der Zigarettenpackung werden Drehungen um zwei Achsen ausgewertet. Die Lage der Schachtel in Abbildung 44 erlaubt eine Änderung der Lautstärke, diese Änderung wird durch die Dose realisiert. Hierbei kommt aufgrund der Vorteile der Form im Vergleich zur Pyramide ein anderer Ansatz und eine andere Metapher zum Einsatz. Die Funktionalität ist hier nicht analog zu einer Fernbedienung, sondern zum Lautstärkenregler an einer

Stereoanlage, also die Funktion eines Potentiometers, wobei auch beim minimalen und beim maximalen Wert noch in die entsprechende Richtung weiter gedreht werden kann, jedoch ohne dass sich der Wert verändert. Aufgrund der Ähnlichkeiten der Formen ist die Auswirkung der Manipulation der Dose schnell und einfach verständlich. Die Dose wird also gedreht und abhängig vom Drehwinkel ändert sich der Wert, in diesen Fall die Lautstärke, wobei eine Linksdrehung zurück bzw. weniger bedeutet und eine Rechtsdrehung entspricht vorwärts bzw. einer Erhöhung des Wertes, bei schnelleren Bewegungen ändern sich die Werte entsprechend schneller. Diese einheitliche Anwendung der Dose in der gesamten Anwendung erleichtert die Steuerung.

Wenn die Schachtel ausgehend von der Position zur Lautstärkenänderung um 90° im Uhrzeigersinn um die z -Achse gedreht wird, kann mit der Dose die Position im aktuellen Titel verändert werden. Somit wird mit der Schachtel wieder die Funktion gewählt und mit der Dose kann der Wert, also die aktuelle Position, verändert werden. Bei einer weiteren Drehung um 90° kann die Playlist ausgewählt werden, somit kann durch Drehung der Dose direkt zur nächsten bzw. vorhergehenden Playlist gesprungen werden. Diese Funktionalität gab es bei der Pyramide nicht, sie konnte jedoch hier einfach hinzugenommen werden, da die Schachtel viele, auch visuell unterscheidbare, Zustände zulässt und das durchführen dieser Gesten sehr einfach möglich ist. Das Wechseln der Playlist kann mit dem Wechseln der CD bei einem CD-Player verglichen werden, wobei die Playlist beliebige Titel beinhalten kann.

Ausgehend von der Playlistwahl führt eine weitere Drehung um 90° im Uhrzeigersinn zur Titelwahl, hierbei kann innerhalb einer Playlist ein Titel gewählt werden. Auch hier gibt es wieder einen Unterschied zur Pyramide, da es hier keine direkte Wahl eines Titels gibt, dies ist jedoch auch nicht unbedingt erforderlich, da durch das schnelle Drehen der Dose sehr schnell um mehrere Titel weiter gesprungen werden kann und somit der gewünschte Titel schnell erreicht wird. Weiters ist diese Form der Titelauswahl etwas intuitiver als die Eingabe der Ziffern für die direkte Titelauswahl.

Neben der Drehung um die z -Achse spielt auch die Drehung um die x -Achse eine Rolle. So bedeutet die Lage in Abbildung 44 neben der Möglichkeit zur Lautstärkenregelung auch, dass sich der Player im Zustand Stopp befindet. Das bedeutet, dass die Drehungen um die z -Achse bestimmen, welche Werte durch die Dose manipuliert werden können, eine Drehung um die x -Achse hingegen beeinflusst den Zustand des Players, die drei möglichen Zustände sind Stopp, Pause und Play. Wird die Schachtel um 90° um die x -Achse gedreht, sodass sie auf der länglichen Seitenfläche liegt, so wechselt der Player vom Zustand Stopp in den Zustand Pause, hierbei ist es egal ob die Drehung in oder gegen den Uhrzeigersinn erfolgt. Im Pause Zustand können die Lautstärke, die Position im Titel, usw., auch wieder geändert werden, wobei die Auswahl identisch mit den zuvor erwähnten Drehungen um die z -Achse erfolgt. Wird die Schachtel vom Zustand Pause um weitere 90° gedreht, so wird der aktuelle Titel abgespielt da der Player in den Zustand Play wechselt. Natürlich können auch hier wieder die Werte, abhängig vom yaw-Wert des Sensors, durch die Manipulation der Dose geändert werden.

Diese Wahl der Zustände erlaubt neben der einfachen Handhabung auch die Möglichkeit den Zustand des Players durch die Schachtel darzustellen. So kann, da bei der Schachtel auch aufgrund des Aufdrucks zwischen Vorder- und Rückseite und auch zwischen oben und unten unterschieden werden kann, durch einen Blick erkannt werden, in welchem Zustand (Stopp, Pause oder Play) sich der Player befindet und zusätzlich ist es möglich, festzustellen welcher der vier Werte (Lautstärke, Position im Titel, Titel und Playlist) aktuell durch die Dose verändert werden kann. Informationen über die Werte, wie etwa die Lautstärke, werden durch die Orientierung der Artefakte jedoch nicht dargestellt.

Generell ist zu sehen, dass die Steuerung eines Players sehr einfach in zwei TUIs integriert werden kann und diese beiden Formen leicht für die grundlegende Funktionalität ausreichen. Dies war bereits bei der Pyramide zu sehen, da hier nur eine Form einen ähnlichen Funktionsumfang abdeckte. Der Vorteil der beiden TUIs, der sich auch durch die Kopplung einer Eingabemöglichkeit für diskrete und für kontinuierliche Werte ergibt, liegt jedoch sicher in der daraus resultierenden intuitiveren und einfacheren Handhabung und der damit verbundenen Akzeptanz beim Benutzer.

Die Anforderung an diese Anwendung die Möglichkeiten der Gestenerkennung darzustellen wurde in beiden Realisierungswegen großteils erfüllt, wenn auch nicht vollständig ausgenutzt. Speziell der Bereich der komplexeren Gesten wurde in keiner der beiden Anwendungen vollständig abgedeckt, da es kaum möglich ist eine große Anzahl von intuitiven und komplexen Bewegungen zu finden die zusätzlich noch einfach und mit einer geringen Fehlerwahrscheinlichkeit wiederholbar sind. Die Abbildung von erkannten Gesten auf eine beliebige Aktion durch den Player ist klar zu erkennen, bei manchen Bewegungen werden auch mehrere Gesten auf eine Aktion abgebildet. Durch die beiden Versionen (Pyramide bzw. Dose & Schachtel) wird auch gut gezeigt, dass die Wahl der Gesten abhängig von der geometrischen Form gewählt wurde.

Bei der Pyramide ist die Integration in alltägliches Objekt nicht erfolgt, diese Version war eher als Versuch gedacht, eine große Anzahl von Gesten die trotzdem leicht unterscheidbar sind, zu realisieren. Auch die Repräsentation der digitalen Information gelingt aufgrund der Form der Pyramide gut.

Die Steuerung mit mehreren TUIs erfüllt die Anforderung an die Integration in alltägliche Objekte vollkommen, da die Elektronik nicht wahrzunehmen ist und es sich um zwei Artefakte handelt die im täglichen Leben sonst auch verwendet werden.

Gesamt gesehen veranschaulicht diese Anwendung sehr gut die prinzipiellen Möglichkeiten der verwendeten Hard- und Software und auch die Vision die hinter TUIs steckt wird sehr gut dargestellt.

7 Fazit und Ausblick

Die Entwicklung des TUIs mit der verwendeten Technologie und Funktionsweise hat sich als viel versprechend herausgestellt, zumal dieser Bereich noch sehr jung ist ergeben sich basierend auf dieser Arbeit entsprechende Möglichkeiten weitere Überlegungen durchzuführen. Aufgrund der unterschiedlichen Disziplinen die hier miteinfließen ergeben sich ebenso viele Teilgebiete auf denen sich Weiterentwicklungen anbieten. Eine grobe Einteilung lässt sich zwischen Hardware und Software treffen, zusätzlich gibt es Bereiche die empirisch untersucht werden können um die Qualität dieses TUIs als Eingabeform zu evaluieren. Nicht zuletzt existiert auch im Bereich der Anwendungen und Szenarien noch ein breites Feld für weitergehende Entwicklungen.

7.1 Weiterführende Arbeit

Die Weiterentwicklung kann von einem TUI zu einem persönlichen TUI führen. Eine mögliche Anwendung ist die Signatur mit dem persönlichen TUI mit Hilfe einer durch den Benutzer festlegbaren Bewegung. Hierbei muss der Benutzer über das TUI und die dazugehörige Bewegung verfügen. Spezifische Eigenschaften beim Durchführen der Geste können erkannt werden und stellen somit ein Sicherheitskriterium dar oder es erfolgt die Kennung des Benutzers durch einen RFID-Tag und die Bestätigung einer Transaktion erfolgt durch eine bestimmte Geste. Weiters könnte ein persönliches TUI an die Eigenschaften des Benutzers angepasst werden und eine Steuerung und Interaktion durch das TUI in unterschiedlichen Umgebungen wäre möglich.

Auch Anwendungen für ältere und behinderte Personen sind denkbar, da die Bedienung mittels TUI intuitiv und somit leicht verständlich ist.

7.1.1 Hardware

Der Ansatz bei der Wahl der Hardware ist zwar richtig, da hiermit die tatsächlichen Anforderungen bestimmt werden können und auch noch Spielraum für weitere Versuche bleibt, gerade deswegen ergibt sich daraus ein breites Spektrum für Änderungen und Erweiterungen in diesen Bereich. Diese Modifikationen beginnen bei kleinen Änderungen und gehen bis zu einer kompletten Neuentwicklung der Hardware für eine einzelne Anwendung mit der Optimierung für diese.

Erweiterungsmöglichkeiten der aktuellen Hardwarekonfiguration ergeben sich durch zusätzliche Sensorik für die Eingabe, weiters sind auch Ausgabemöglichkeiten direkt beim TUI von Interesse. Die Anforderungen an eine erweiterte Eingabemöglichkeit hängen von den entsprechenden Anwendungen ab. Die logisch gesehen einfachste Erweiterung ist ein binärer Schalter oder Taster. Hierbei muss es sich nicht um Taster im herkömmlichen Sinn, und zwar ein kleiner Knopf auf den man drücken kann, handeln, sondern er kann durchaus in das Artefakt integriert werden, indem beispielsweise das Berühren einer Fläche eines Würfels als Tastendruck erkannt wird, oder auch das erkannt wird wenn der Würfel auf einer bestimmten Seite aufliegt bzw. etwas darauf liegt. Fast alle Erweiterungen führen eine verhältnismäßig große Änderung der Hardware nach sich. Dies ist auch der Grund, warum diese aktuell noch nicht zur Verfügung stehen. Der erhöhte Aufwand liegt in der Datenübertragung. Derzeit wird der Datenstrom des Sensors direkt an die Auswerteeinrichtung übermittelt. Wenn nun eine weitere Signalquelle, wie etwa der Taster, miteinbezogen werden soll ist ein Multiplexer erforderlich, dieser kann beispielsweise durch einen Mikrocontroller realisiert werden. Der Multiplexer kann die Daten die zum/vom Sensor kommen von anderen Datenströmen, etwa der Eingabe durch einen Taster oder einer Ausgabe für eine LED trennen.

Mit dem Taster könnte man auch die Gestenerkennung erweitern. Bei der Unterscheidung zwischen Gesten mit fixen und variablen Startpunkt (siehe Kapitel 5.8.4) kann der Wert des Tasters als Entscheidungskriterium gewählt werden, auch wenn die Steuerung mittels diesem TUI dann nicht rein intuitiv zu bedienen ist, sondern zumindest ein Grundwissen über die Funktion und die Gesten erfordert.

Neben einem Taster steht eine Vielzahl von weiteren Eingabemöglichkeiten zur Verfügung, für die ein ähnlicher Aufwand für die Adaptierung der Hardware erforderlich ist, auch wenn mit einem Mikrocontroller mehrere solcher Erweiterungen auf einmal erfolgen können. Ein Touchpad könnte auch als zusätzliche Eingabeform verwendet werden, wobei hier die berührungssensitive Oberfläche nicht auf eine Seite beschränkt bleiben muss, sondern sehr wohl alle Seiten umfassen kann. Die normalerweise üblichen Touchpads liefern nur einen Punkt der gedrückt wird. Diese sind somit nur eingeschränkt für diese Anwendung verwendbar, da es beispielsweise von Interesse sein kann, wie das Artefakt in der Hand gehalten wird, also auch mit wie vielen Fingern es gehalten wird. Dies kann nur mit speziellen Touchpads erreicht werden, die multiple simultane Druckpunkte erkennen können. Diese Technologie kann man auch dafür verwenden um einen zusätzlichen bewussten Eingabekanal zur Verfügung zu haben, indem man eigene Anweisungen durch das Zeichnen auf diesen Touchpad ausführen kann. Durch eine zusätzliche Messvorrichtung für die absolute oder zumindest relative Positionsbestimmung könnte auch die Qualität und die Möglichkeiten der Gestenerkennung erweitert werden. Weitere Sensorik kann die Temperatur, Helligkeit, Geräuschpegel oder Ähnliches, abhängig von der genauen gewünschten Anwendung, erfassen. Durch einen Mikrocontroller kann das System somit auf ein Sensorboard zur umfangreichen Kontextbestimmung erweitert werden.

Ähnliche Möglichkeiten gibt es bei der Betrachtung der Ausgabemöglichkeiten direkt am Artefakt, dadurch kann die Usability verbessert und weitere Kommunikationskanäle verwendet werden. Neben der bereits erwähnten LED können wesentlich umfangreichere Anzeigen integriert werden, sodass die Oberfläche des Artefaktes aktuelle Informationen anzeigt. Bei einem Würfel könnte beispielsweise die Funktion der einzelnen Seiten dargestellt werden. Eine weitere Möglichkeit ist die farbliche Gestaltung des Artefaktes abhängig vom Zustand der mittels der Sensorik aufgenommen wird, so kann das Erscheinungsbild des TUIs an die Situation angepasst werden.

Durch eine Integration von force Feedback kann der Benutzer direkt über die Gestenerkennung informiert werden, also etwa wenn eine Geste erfolgreich erkannt wurde. Hierbei muss jedoch darauf geachtet werden das die Sensorwerte durch die Bewegungen nicht verfälscht werden.

Eine weitere Möglichkeit die auch den Hardwarebereich betrifft ist die Integration in Kleidungsstücken oder generell in Artefakte die am Körper getragen werden, hierbei können auch mehrere Sensoren an unterschiedlichen Stellen (Arme, Beine, Kopf,...) verwendet werden um ein besseres Gesamtbild der Bewegungen zu erhalten. Prinzipiell, könnte die Sensorik auch in Gegenstände wie eine Uhr, ein Halsband, die Schuhe usw. integriert werden. Dadurch erhält man ein sehr umfangreiches Bild der Bewegungen, ohne dass sich der Benutzer um zusätzliche Befestigungsmöglichkeiten der Sensoren kümmern muss.

Weiters ist die Überlegung nach einer komplett anderen Sensorik nicht zu verwerfen, wenn es um einen längeren Betrieb oder eine Anpassung für eine spezielle Anwendung mit geringeren Anforderungen oder dem Hauptaugenmerk auf einen bestimmten Bereich geht.

7.1.2 Software

Der Softwareteil geht meist Hand in Hand mit den Änderungen an der Hardware, somit ergeben sich aus den Modifikationen im vorhergehenden Punkt auch immer entsprechende Weiterentwicklungen der Software, diese werden hier jedoch nicht extra behandelt, da hierbei nähere Details über die Änderungen der Hardware erforderlich sind.

Eine wesentliche Weiterentwicklung der Gestenerkennung kann durch die Erweiterung auf mehrere Sensoren erfolgen, wobei hier nicht unbedingt nur Orientierungssensoren gemeint sind. Durch die Betrachtung mehrerer Messwerte - beispielsweise von einem Sensor auf der linken und einem auf der rechten Hand, können umfangreichere Gesten erkannt werden oder es kann eine genauere Aussage über den Kontext getroffen werden, in dem sich der Benutzer oder ein Artefakt mit dem Sensor befindet.

Die Erkennung der Gesten aufgrund der Sensordaten kann durch verbesserte oder möglicherweise auch komplett neue Algorithmen erhöht werden, da bis jetzt das Hauptaugenmerk nicht darauf lag und entsprechend einfache Algorithmen eingesetzt werden die es erlauben, die Gestenerkennung anzupassen, also etwa ob eine höhere Erkennungswahrscheinlichkeit gewünscht wird, oder beispielsweise ob das Hauptaugenmerk darin liegt, keine Gesten falsch zu erkennen. Weiters kann eine Erweiterung des Gestenvorrates in die Richtung eines umfangreichen Gestenalphabetes, welches durch das API zur Verfügung gestellt wird, sinnvoll sein, um somit die Entwicklung von Anwendungen zu erleichtern.

7.1.3 Empirische Untersuchungen

Auf dem Bereich der Benutzerstudien gibt es noch einige Punkte die durchgeführt werden sollen. Die erste Untersuchung betrifft einen grundlegenden Bereich, und zwar, wie schnell, einfach, fehleranfällig, benutzerfreundlich, usw. die Eingabe von Gesten erfolgen kann. Hierzu kann ein Gestenvorrat verwendet werden um die Ergebnisse mit anderen Eingabeformen vergleichen zu können, auch wenn zentrale Kriterien für diese Anwendung wie „Natürlichkeit“ kaum objektiv zu evaluieren sind. Es kann hiermit auch eine Bewertung der einzelnen Gesten eines Gestenalphabetes vorgenommen werden.

Ein etwas greifbarer Vergleich kann bei der Steuerung des Mediaplayers gefunden werden. Hier können einerseits die beiden Realisierungen (Pyramide bzw. Dose & Schachtel) verglichen werden, andererseits kann auch die konventionelle Steuerung mittels GUI oder eine Steuerung mittels Fernbedienung in den Vergleich miteinbezogen werden. Dadurch kann eine qualitative Aussage über die Bedienbarkeit der Anwendung und über die Vorteile der unterschiedlichen Ansätze getroffen werden.

Eine weitere Untersuchung betrifft die verschiedenen geometrischen Formen und die Eignung dieser für die unterschiedlichen Anwendungen. Bisher wurden nur Formen aus Acrylglas und zwei Formen aus dem Alltag (Dose und Schachtel) verwendet und die Betrachtung beschränkte sich auf ein paar geometrische Formen, somit kann noch keine Aussage über die Vor- und Nachteile verschiedener Materialien, Oberflächen, Gewichtsverteilungen, Farbe, usw. oder auch über andere unregelmäßige Körper gemacht werden.

7.2 Fazit

Einer der grundlegenden Punkte, die Machbarkeit, konnte durch die Anwendungen sehr anschaulich gezeigt werden, es ist auch erkennbar, dass man mit TUIs der Hand die volle Handlungs- und Ausdruckskraft im Umgang mit der Umgebung geben kann.

Bei der Realisierung war die Entscheidung für diese Sensorklasse und auch die Art der Datenübertragung angebracht. Obwohl die Sensorik nur Informationen über die Orientierung

liefert, kann eine große Anzahl von Gesten erkannt werden, ohne dass eine zusätzliche Infrastruktur benötigt wird.

Die Anwendungen, allen voran der Mediaplayer, veranschaulichen die Möglichkeiten des Frameworks und der Hardware sehr gut. Es ist sehr gut ersichtlich wie die Elektronik in einen alltäglichen Gegenstand (Zigaretenschachtel, Getränkedose, ...) integriert werden kann und daraus ein smartes Artefakt entsteht.

Durch die beliebige Zuordnung der erkannten Gesten auf Aktionen der Anwendungen kann die Bewegungserkennung sehr flexibel eingesetzt und an den Benutzer angepasst werden. Die Konfigurier- und Adaptierbarkeit wird nicht zuletzt durch die freie Gestaltung und Verwaltung der Gesten zur Laufzeit gewährleistet. Nicht zuletzt aufgrund der geringen Abhängigkeit des Frameworks von der Hardware ist die Verwendung einfach möglich, dies betrifft auch die Integration der Gestenerkennung in eine Anwendung.

Eine weitere wesentliche Eigenschaft ist, dass die Gesten auch unabhängig von der Orientierung des Sensors wahrgenommen werden können.

Zwei Einschränkungen ergeben sich aufgrund der Wahl des Sensors, diese können jedoch durch die Wahl eines anderen Sensors verhindert werden. Der erste Punkt ist die Beeinflussung der Sensorwerte durch einen metallischen Gegenstand in unmittelbarer Nähe zum Sensor und der zweite Punkt ist das Verhalten des Sensors bei einem bestimmten Sensorwert. Wenn der pitch-Wert eine gewisse Grenze überschritten hat, hat dies Einfluss auf die beiden anderen Sensorwerte, diese Nebenwirkung kann jedoch nicht garantiert von einer entsprechenden Rotation des Sensors unterschieden werden und es kann somit keine eindeutige Aussage bei einer Rotation des Sensors in diesen Bereich über die beiden anderen Sensorwerte gemacht werden.

Es wurden keine empirischen Studien durchgeführt um die Usability zu testen, dies ist erforderlich um eine umfangreichere Beurteilung über die Eignung dieser TUIs abgeben zu können. Auch eine Bewertung der Natürlichkeit aller verwendeten Gesten ist ausständig.

Gesamt gesehen kann dieser Prototyp als guter Ansatz und als Basis für weitere Entwicklungen auf diesem Gebiet gewertet werden, auch wenn derzeit noch nicht alle Möglichkeiten ausgeschöpft werden.

Literatur

- [1] BeeCon, Spezifikation BlueWand IIx, <http://www.beecon.de>
- [2] Benbasat, A. Y., Paradiso, J. A., An Inertial Measurement Framework for Gesture Recognition and Applications, Gesture and Sign Language in Human-Computer Interaction, 2001 Proceedings, Springer-Verlag, Berlin, 2002
- [3] Bluetooth SIG, Bluetooth Core Specification, Revision 1.2, November 2003
- [4] Brauer, V., Feature-basierte Erkennung dynamischer Gesten mit einem Datenhandschuh, Diplomarbeit, Universität Bremen, November 1994
- [5] Bundesamt für Sicherheit in der Informationstechnik, Drahtlose lokale Kommunikationssysteme und ihre Sicherheitsaspekte, 2003
- [6] connectBlue, OEM Serial Port Adapter, Electrical & Mechanical Datasheet, Version 1.5, Dokument Nr. cBProduct-0212-01, Mai 2003
- [7] connectBlue, OEM Serial Port Adapter, Spezifikation, <http://www.connectblue.se/>
- [8] Craig, W. C., Zigbee: "Wireless Control That Simply Works", White Paper, ZMD America, Inc., 2004
- [9] Crampton Smith, G. The hand that Rocks the Cradle, ID Magazine, May/June 1995
- [10] Davis, J., Shah, M., Visual Gesture Recognition, IEEE Vision, Image and Signal Processing, University of Central Florida, Orlando, April 1994
- [11] Fels, S. S., Hinton, G. E., Glove-TalkII: A neuronal network interface which maps gestures to parallel format speech synthesizer controls, IEEE Transactions on Neuronal Networks, University of Toronto, September 1998
- [12] Fitzmaurice, G. W., Graspable User Interfaces, Dissertation, Toronto, 1996
- [13] Freeman, W.T., Roth, M., Orientation Histograms for Hand Gesture Recognition, Intl. IEEE Workshop on Automatic Face and Gesture Recognition, Zürich, Juni 1995
- [14] Frenzel, L., Supplement to Electronic Design, Electronic Design's Special Report, The ZigBee Buzz Is Growing: New Low-Power Wireless Standard Opens Powerful Possibilities, January 2004
- [15] Hamette, P., Lukowicz, P., Tröster, G., Svoboda, T., Fingermouse: A Wearable Hand Tracking System, UBICOMP Conference Proceeding, ETH Zürich, September 2002
- [16] Heinz, E. A., Kunze, K. S., Sulisty, Junker, H., Lukowicz, P. Tröster, G., Experimental Evaluation of Variations in Primary Features Used for Accelerometric Context Recognition, EUSAI2003, November 2003
- [17] Hollar, S., Pister, K. S. J., Wireless Static Hand Gesture Recognition with Accelerometers – The Acceleration Sensing Glove, Technical Report, University of California, Berkley, June 2000
- [18] Hornecker, E., Tangible User Interface als kooperationsunterstützendes Medium, Dissertation, Universität Bremen, April 2004
- [19] Infrared Data Association, Homepage <http://www.irda.org/>
- [20] InterSense InertiaCube² Handbuch, Dokument Nr. 072-IC210-0J01, Revision 1.1, 2001
- [21] InterSense InertiaCube² Spezifikation, <http://www.isense.com>

- [22] Ishii, H., Ullmer, B., Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, ACM Proceedings of CHI97, March 1997
- [23] Java3D Tutorial, <http://raphaello.univ-fcomte.fr/IG/Java3D/Java3D.htm>
- [24] Kinney, P., ZigBee Technology: Wireless Control that Simply Works, White Paper, October 2003
- [25] Klein, M., Harbaum, T., Fuhrmann, T., Bluetooth-Artefakte zur Interaktion mit intelligenten Umgebungen, Szwillus, G., Ziegler (Hrsg.), J., Mensch & Computer 2003: Interaktion in Bewegung, Stuttgart: B.G. Teubner, 2003
- [26] Kohler, M., Schröter, S., Handgestenerkennung durch Computersehen, Informatik'98 (Dassow, J., Kruse (Hrsg.), R., Springer-Verlag Seite 201-212), Universität Dortmund, September 1998
- [27] LaViola, J.J., A Survey of Hand Posture and Gesture Recognition Techniques and Technology, Technical Report CS99-11, Brown University, Rhode Island, June 1999
- [28] Mäntyjärvi, J., Himberg, J., Seppänen, T., Recognizing Human Motion with multiple Acceleration Sensors, Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, June 2001
- [29] Maschtera, W., Wireless LANs, Vorlesungsunterlagen, JKU Linz, SS 2003
- [30] Mattern, F., Ubiquitous Computing, Vorlesungsunterlagen, JKU Linz, SS2002
- [31] Mattern, F., Ubiquitous Computing, Informatik – Zeitschrift der schweizerischen Informatikorganisation, Oktober 2001
- [32] McGee, D. R., Cohen, P. R., Wu, L., Something from nothing: Augmenting a Paper based work practice via multimodal interaction, ACM Proceedings of Designing Augmented Reality Environments, Helsingor, Denmark, April 2000
- [33] Milgram, P., Takemura H., utsumi, A., Kishino F., Augmented Reality: A class of displays on the reality-virtuality continuum, Proceedings of SPIE, Telemanipulator and Telepresence Technologies, 1994
- [34] Moys remote control Plugin, <http://www.winamp.com/plugins/details.php?id=64427>
- [35] National Semiconductor, LM2940 - 1A Low Dropout Regulator, Datenblatt, July 2000
- [36] Niethard, F. U., Pfeil, J., Orthopädie (4. Auflage), ISBN: 3131308141, Thieme Verlag, Stuttgart, April 2003
- [37] Nölker, C., GREFIT: Ein System zur Visuellen Erkennung von Handposturen, Doktorarbeit, Bielefeld, Oktober 2000
- [38] Pering, T., iCube: a Tangible Interface for Mobile Interaction, Toolkit Support for Interaction in the Physical World, Pervasive 2004 Workshop, April 2004
- [39] Perng, J. K., Fisher, P., Hollar, S., Pister, K. S. J., Acceleration Sensing Glove (ASG), ISWC International Symposium on Wearable Computers, University of California, Berkley, October 2000
- [40] Piper, B., Ratti, C., Ishii, H., Illuminating Clay: A 3-D Tangible Iternface for Landscape Analysis, Proceedings of CHI 2002, April 2002
- [41] Piper, B., Ratti, C., Ishii, H., Illuminating Clay: A Tangible Interface with potential GRASS applications, Proceedings of the Open Source GIS – GRASS user conference 2002, September 2002
- [42] Radiometrix, Homepage <http://www.radiometrix.co.uk/>

- [43] Rekimoto, J., Sciammarella, E., ToolStone: Effective Use of the Physical Manipulation Vocabularies of Input Devices, ACM Proceedings of UIST2000, November 2000
- [44] Schmidt, A., Ubiquitous Computing – Computing in Context, Dissertation, Lancaster University, November 2002
- [45] Schürmann, J., Einführung in die Hochfrequenz Identifikations Technologie, GME-Fachbericht Nr. 13, Identifikationssysteme und kontaktlose Chipkarte, VDE-Verlag, Berlin 1994
- [46] Starner, T. E., Visual Recognition of American Sign Language Using Hidden Markov Models, Master's thesis, Massachusetts Institute of Technology, February 1995
- [47] Starner, T. E., Weaver, J., Pentland, A., A Wearable Computer Based American Sign Language Recognizer, IEEE International Symposium on Wearable Computers, Massachusetts Institute of Technology, October 1997
- [48] Sturman, D. J., Zeltzer, D., A Survey of Glove-based Input. IEEE Computer Graphics and Application, 1994
- [49] Sturman, D. J., Whole-Hand Input, Ph. D. Thesis, Massachusetts Institute of Technology, February 1992
- [50] Ullmer, B., Ishii, H., Glas, D., mediaBlocks: Physical Containers, Transports, and Controls for Online Media, SIGGRAPH98, July 1998
- [51] Van Laerhoven, K., Cakmakci, O., What Shall We Teach Our Pants?, Proceedings of International Conference on Wearable Computing 2000, October 2000
- [52] Van Laerhoven, K., Villar, N., Schmidt, A., Kortuem, G., Gellersen, H., Using an Autonomous Cube for Basic Navigation Input, Proceedings of ICMI/PUI 2003, November 2003
- [53] Watson, R., A Survey of Gesture Recognition Techniques, Technical Report TCD-CS-93-11, Trinity College, Dublin, July 1993
- [54] Weiser, M., The Computer for the 21st Century, Scientific American, September 1991
- [55] Weiser, M., Brown, J. S., Designing Calm Technology, Xerox PARC, <http://www.ubiq.com/weiser/calmtech/calmtech.htm>, December 1995
- [56] Wellner, P., Interacting with Paper on the DigitalDesk, Communications of the ACM, July 1993
- [57] Wellner, P., The DigitalDesk Calculator: Tangible manipulation on a Desk Top Display, ACM Proceedings of UIST91, November 1991
- [58] Winamp Player Homepage, <http://www.winamp.com/>
- [59] ZigBee™ Alliance, Homepage: <http://www.zigbee.org/>

Lebenslauf

Der Lebenslauf wurde in der Online Version entfernt, kann jedoch auf Anforderung zugesandt werden.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wartberg ob der Aist, am 29.10.2004.