



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Haci Yigit

Entwicklung eines gestenbasierten Steuerungssystems
für eine Smart Home Umgebung

Fakultät Design, Medien und Informatik
Department Medientechnik
Studiengang Media Systems

Faculty of Design, Media and Information
Department Media Technology
Media Systems

Haci Yigit

Entwicklung eines gestenbasierten Steuerungssystems
für eine Smart Home Umgebung

Haci Yigit

Matrikelnummer: 2059113

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Media Systems

am Department Medientechnik

der Fakultät Design, Medien und Information

der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Prof. Dr. Roland Greule

Zweitprüfer: Prof. Dr. Andreas Plaß

Eingereicht am 02.Februar.2016

Haci Yigit

Thema der Arbeit

Entwicklung eines gestenbasierten Steuerungssystems für eine Smart Home Umgebung

Stichwörter

Smart Home, Gestenerkennung, Kinect, Arduino

Zusammenfassung

Die Entwicklung eines gestenbasierten Steuerungssystems für eine Smart Home Umgebung wird in dieser Bachelorarbeit thematisiert. Die Konzeption und die Realisierung der Gestenerkennung sind die Schwerpunkte dieses wissenschaftlichen Beitrags. In Rahmen dieser Arbeit wurden Anforderungen definiert, um möglichst ein Smart Home konformes System zu realisieren. Die Thesis umfasst dabei die praktische Umsetzung der konzeptionellen Ausarbeitung, um die Interaktionen zwischen dem Menschen und dem Computer zu ermöglichen.

Haci Yigit

Title of the paper

Development of a gesture-based control system for a smart home environment

Keywords

Smart Home, gesture recognition, Kinect, Arduino, LED

Abstract

The development of a gesture-based control system for a smart home environment is demonstrated in this work. The conception and the realization of the gesture recognition are the focus of this scientific contribution. In this work conditions in order to realize potential smart home compliant system defined. The work consists of the practical implementation of the conceptual development and thus enables the interactions between the human and the computer.

Danksagung

Ich bedanke mich bei meinem Betreuer Prof. Dr. Roland Greule, der sehr geduldig war und mich stets unterstützt hat. Zudem möchte ich mich auch bei Herrn Prof. Dr. Pläß für die Ratschläge bedanken. Ein großes Dankeschön ist an meine Familie gerichtet, die stets an meiner Seite war.

Inhaltsverzeichnis

1 Einleitung	8
1.1 Motivation	8
1.2 Ziel der Arbeit	9
1.3 Gliederung der Arbeit	9
2 Grundlagen	9
2.1 Gesten	10
2.2 Motion Capture	12
2.2.1 Optisches Tracking	12
2.3 Human-Computer-Interaction	13
2.4 Ubiquitous Computing	14
2.5 Smart Home	15
3. Analyse	17
3.1 Konzept zur Steuerung	17
3.2 Eingabealphabet in 2D	17
3.3 Unistroke	18
3.4 Start-Stopp-Problematik	19
3.5 \$1 unistroke recognize	22
3.6 Konzeptionelle Gestenanforderungen	23
3.6.3 Regulierung der Parameter	26
3.6.4 Das Auswählen	28
3.7 Szenarien	29
4. Umgebung	30
4.1. Einsatzgebiete	30
4.2 Hardware	30
4.2.1 Kinect Xbox 360	31
4.2.2 Arduino Due	32
4.2.3 LED Strip (WS2812B)	33
4.3 Systementwurf	35
4.4 Software und Sprache	35
4.4.1 WPF Framework	35
4.4.2 Processing	36
4.4.3 Windows Developer Toolkit v1.8	36
5 Design	36
5.1 Programmablauf	36
5.2 Erweiterung des Systems	38
6. Realisierung	38

6.1 Dynamic-Time-Warping Algorithmus	38
6.2 Kinect Code.....	40
6.3 Processing und Pololu Lib	43
6.5 Einblick.....	44
7 Schluss	45
7.1 Zusammenfassung.....	45
7.2 Ausblick.....	46
Literaturhinweis	47

1 Einleitung

In einer entwickelten Gesellschaft ist der technische Fortschritt von essentieller Bedeutung, insbesondere die der Informationstechnik. Der Computer ist aus dem Leben des Menschen nicht mehr weg zu denken. Die Anzahl und die Komplexität der Computer sind enorm gestiegen. Dabei ist die Steuerung der Computer durch die Peripheriegeräte nahezu gleich geblieben, wenn man die Touch-Technologie nicht in Betracht zieht. Die Eingabeform der Informationen von Mensch zu Computer entwickelte sich von mechanischen Schaltern (bzw. Tasten), die immer noch stark verbreitet sind, zum berührungsempfindlichen Touchscreen. Im Zeitalter der Ubiquitous Computing ist die Nachfrage nach einer praktischen Steuerungsmöglichkeit, die möglichst viele Systeme abdeckt und damit viele Eingabegeräte ersetzt, groß. Eine passende Steuerungsmöglichkeit ist die Gestensteuerung, welche in dieser Arbeit näher untersucht wird. Der Fokus dieses wissenschaftlichen Beitrags ist auf die Konzeption und Entwicklung des Gestenerkennungssystems für eine Smart Home Umgebung gelegt, um aufzuzeigen, dass diese Steuerungsmöglichkeit eine Alternative darstellt. Die Fragestellung der Wichtigkeit der Gestenerkennung in der Zukunft bezüglich Mensch-Maschine Interaktion ist ein interessantes Themengebiet für die Forschung als auch für Industrie.

“I believe we will look back on 2010 as the year we expanded beyond the mouse and keyboard and started incorporating more natural forms of interaction such as touch, speech, gestures, handwriting, and vision— what computer scientists call the ‘NUI’ or natural user interface.”
[Ballmer, 2010]

1.1 Motivation

Es ist zu beobachten, dass die Steuerung via Gestenerkennung längst in den Branchen wie Game, Auto, Entertainment, Heimeräte und Industrie längst präsent ist, jedoch das Potenzial noch lange nicht ausgeschöpft ist. Eine Steuerungsmöglichkeit, die praktisch ist und zudem viele Geräte abdeckt, ist sehr gefragt. Beispielsweise arbeiten große Autohersteller wie Volkswagen an Gestensteuerung, um den Bordcomputer anzusprechen. „Gestensteuerung ist Schlüssel zur Zukunft“[1], so beschreibt der Autohersteller die Zukunft der Automobilbranche auf der eigenen Seite. Für den Heimgebrauch bieten verschiedene Fernsehgeräte derweil Modelle an, die auch per Geste gesteuert werden können. In der Entertainmentbranche ist auch die Zeit der Virtual Reality (VR) angebrochen, und da ist die Nachfrage nach einem System, welches ohne ein unpraktisches Eingabegerät auskommt, groß. Die hardwarelose Bedienung, also die nicht getragen oder berührt werden muss, diese Art der Steuerung, die bei einer Gestensteuerung gegeben ist, ist hingegen vorteilhaft gegenüber den gängigen Peripheriegeräten wie Fernbedienung oder Controllern. In einer vernetzten Welt ist die Eingabe von Informationen durch Geste eine gute Alternative.

1.2 Ziel der Arbeit

Ziel der Arbeit ist einen Prototypen für ein Smart Home Environment zu konzipieren und zu entwickeln, welcher Grundlage der Steuerung der verschiedenen Geräte wie Musik-Anlage, Fernseher und Beleuchtung bietet, um eine praktische Human Computer Interaction (HCI) zu ermöglichen. Ein Ziel ist auch unter anderem ein Repertoire an Gesten (Gestenalphabet) zu konzipieren, welches möglichst durch vordefinierte Kriterien Smart Home konform ist und diese anschließend umzusetzen. In dem praktischen Teil der Arbeit wird die Entwicklung der Gestenerkennung als auch die Programmierung unter Berücksichtigung des Konzepts "Ein Buchstabe, eine Komponente" gezeigt. Im praktischen und schriftlichen Teil der Arbeit ist die Steuerung der LED-Beleuchtung als Beispiel ausgewählt worden, um die Funktionsweise des Systems zu zeigen. Das System kann mit unterschiedlichen Komponenten erweitert werden, die unter anderem in den Bereichen Licht, Musik, Bild und Heimgeräte zu finden sind. Im Rahmen dieser wissenschaftlichen Arbeit wird auf die Entwicklung der Gestenerkennung eingegangen. Eine angepasste Variation des Dynamic-Time-Warping Algorithmus kommt zum Einsatz, um die Gestenerkennung zu ermöglichen.

1.3 Gliederung der Arbeit

Die Thesis ist in sieben Kapiteln unterteilt. Alle folgenden Kapitel bauen aufeinander auf. Im zweiten Kapitel werden die Grundlagen erläutert, um wichtige Kernpunkte zu erläutern. Das dritte Kapitel beschreibt die Analyse der Arbeit. Es wird dabei auf die Konzeption der Gesten eingegangen und die Kriterien der Gesten aufgezeigt. Im nächsten Kapitel wird die Umgebung des Systems erläutert. Dabei werden die Hard- und Softwarekomponenten beschrieben, die zum Einsatz kommen. Der endgültige Aufbau wird in Kapitel fünf beschrieben. Die Übersicht des Systems ist abgebildet. Auf die praktische Realisierung wird im vorletzten Kapitel eingegangen. Es werden wichtige Programmfunktionen aufgezeigt und die Funktionsweise der Gestenerkennung konkret erläutert. Eine Bewertung dieses wissenschaftlichen Beitrags ist im letzten Kapitel zu finden.

2 Grundlagen

In diesem Kapitel werden die Grundlagen erläutert, die für das Verständnis dieser Arbeit notwendig sind. Themen wie human-computer interaction (HCI), Ubiquitous Computing werden aufgezeigt, die den Beweggrund dieser Arbeit darstellen. Im ersten Abschnitt des Kapitels wird tiefgründig auf die Fragestellung „Was ist eine Geste?“ eingegangen. Außerdem werden die Technologien, die für die Erkennung der Gesten zum Einsatz kommen, aufgezeigt. In den weiteren Abschnitten wird unser Zeitalter, das des Ubiquitous Computing, erläutert und die Interaktion zwischen Mensch und Computer in der Informatik wissenschaftlich definiert.

2.1 Gesten

Eine Art der non-verbalen Kommunikation ist die Geste, die unter den Menschen bewusst oder unbewusst mitgeteilt wird, um Emotionen oder Informationen wie in der Gebärdensprache mitzuteilen. Streng genommen ist eine Geste die Bewegung bzw. Position der Gliedmaßen von Hand, Finger und dem Gesicht. Das Wort Geste kommt vom lateinischen Wort *gestus*, was Gebärde oder Verhalten bedeutet. Es gibt keine eindeutige Definition von Geste. In der Encyclopedia Americana ist eine Geste als *"Jede bewusste oder unbewusste Körperbewegung, außer den Vokalisierungsbewegungen, durch die wir entweder mit uns selbst oder mit anderen kommunizieren."* (Encyclopedia, 2003) definiert.

Ein und dieselbe Geste kann in verschiedenen Kontexten unterschiedliche Bedeutungen haben. Für die Bedeutung der Gesten ist außerdem die Kultur, die Zeit und die Umgebung entscheidend. Eine Spezifizierung der Gesten macht Pavlovic (Pavlovic u.a. 1997) und kategorisiert sie unter kommunikative und manipulative Gesten. Einige konkrete Beispiele zu kommunikativen und manipulativen Gesten sind unten aufgeführt.

Beispiel 1 Das Lachen

Lachen ist eine kommunikative Geste. Die Positionierung der Wangen, vom Mund und den Augenbrauen sind entscheidend für die Geste. Die Töne, die beim Lachen entstehen, unterstreichen die Geste, sind aber kein Ausschlusskriterium. Eine Abfolge von Bewegungen ist nicht zwingend notwendig, um die Geste Lachen zu erkennen, wie es in der Abbildung 1 der Fall ist. Bei der Kommunikation vermittelt das Lachen die emotionale Stimmung. Die Gelotologie ist die Wissenschaft der Auswirkungen des Lachens.



Abbildung 1: Geste lachen bei Menschen

Quelle: [Yuri Arcurs 2009]

Beispiel 2 Das Winken

Die horizontale Bewegung in Richtung Rechts und Links der Hand in etwa der Höhe des Gesichts ist das Winken. Im sozialen Umfeld wird diese Geste eingeleitet, wenn Menschen sich begrüßen und Abschied nehmen. Die non-verbale Geste dient zur Kommunikation unter den Menschen. Eine Abfolge von Bewegungen ist notwendig, um die Geste als Winken zu interpretieren. Das Ausrichten der Handfläche ohne die horizontale Bewegung kann nämlich als Stopp- oder Warnzeichen gedeutet werden.



Abbildung 2: Winken

Beispiel 3 Die Drehung

Die Geste Drehung ist eine manipulative Geste. Diese Bewegung wird bei Geräten durchgeführt, um bestimmte Einstellungen vorzunehmen. In der Regel werden dazu die Regler eines Geräts benutzt. Diese Regler sind bei Musikanlagen und auch bei Beleuchtungen vorzufinden. Die Konvention ist, dass die Drehung nach rechts die Einstellung erhöht bzw. verstärkt und nach links verringert.



Abbildung 3: Regler einer Musikanlage

Quelle: [Pixabay.com 2015]

In dieser Arbeit sind die manipulativen Gesten relevant. Für den theoretischen als auch den praktischen Teil der Arbeit werden die zweidimensionalen Gesten in Betracht gezogen. Dabei sollen die Gesten ohne Marker anderen Objekten eingeleitet werden können. Der Anwender soll durch die Bewegung der Hände die Komponenten in einer Smart Home Umgebung steuern können. Im Kapitel 3.6.2 werden die manipulativen Gesten, die in dieser Arbeit zum Einsatz kommen, erläutert.

2.2 Motion Capture

Der Begriff Motion Capture setzt sich aus dem lateinischen Wort *motion* und dem englischen Begriff *capture* zusammen. Ersteres bedeutet Bewegung und letzteres erfassen. Ein Motion Capture System erfasst komplexe Bewegungsabläufe eines Anwenders und speichert bzw. leitet die Bewegungsdaten weiter. Es gibt verschiedene Systeme bzw. Technologien, um die Bewegungen aufzunehmen. Je nach System können verschiedene Körperbereiche eines Anwenders „getracked“ werden. Gängige Varianten sind unter anderem Eyetracking, Headtracking. Die gängigen Technologien, die bei Motion Capture in Einsatz kommen, sind die Inertiale (Bewegungssensoren), elektromechanische und optische Systeme. Der Einsatz der unterschiedlichen Technologien hängt von den Voraussetzungen, Umgebungen und dem Budget ab. Die verschiedenen Systeme haben unterschiedliche Vor- und Nachteile. Oft wird Motion Capture nur mit Marker basierten Systemen assoziiert, was nicht der Fall ist. Vielmehr ist es ein Teilbereich der Trackingsysteme.

2.2.1 Optisches Tracking

Die optischen Systeme sind in zwei Varianten zu unterteilen. Man unterscheidet zwischen Marker basierten Tracking Systeme und nicht Marker basierten Systemen. Bei Marker basierten Systemen wird der eigentliche Körper nicht erfasst, sondern die Marker, was die Auswertung der Bewegungen angeht. Sie markieren bestimmte Glieder des Anwenders und geben Aufschluss darüber, zu welchem Zeitpunkt sich der Marker an welchem Ort exakt befunden hat. Das System ist auf ein paar Millimeter genau. Komplizierte Bewegungsabläufe können analysiert und auf ein bereits modelliertes Objekt zugewiesen werden, was bei der Erstellung von Filmen und Spielen bereits Praxis ist.



Abbildung 4: MoCap (motion capture)

Quelle: [XSense.com]

Auf der linken Seite sind die Marker abgebildet und rechts das Model. Bei nicht Marker basierten Trackingsystemen wird der komplexe Bewegungsablauf des Anwenders durch Kameras erfasst. Das Rohmaterial sind Videoaufnahmen, aus denen noch keine Informationen über die Positionen der Gliedmaßen hervorgehen. Komplizierte Algorithmen können anhand der Silhouette und der Bewegungen des Anwenders die Glieder durch aufwendige Verfahren erfassen und so die Position bestimmen. Der große Vorteil des Systems ist, dass keine speziellen Anzüge oder Marker getragen werden müssen.

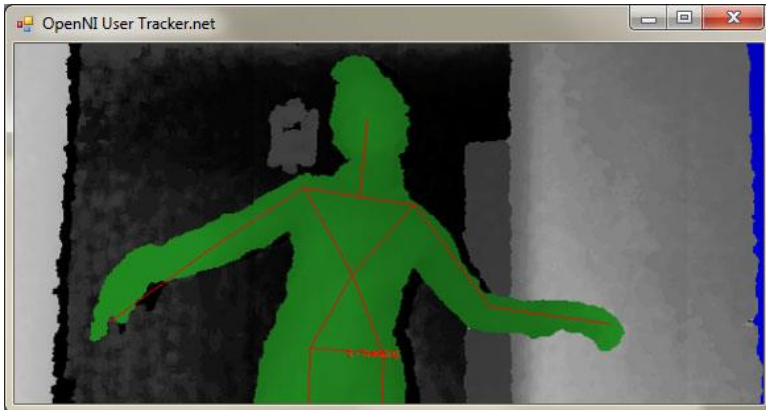


Abbildung 5: Getting the raw and depth image using OpenNI

Quelle: [Codeproject Vangos Pterneas 2013]

Der Anwender wird ohne Marker erfasst.

2.3 Human-Computer-Interaction

Ein Bereich der Informatik ist die Human Computer Interaction (HCI), welche sich mit der Interaktion zwischen Menschen und Computer befasst. Die Interaktion zwischen Mensch und Computer ist allgegenwärtig im Zeitalter des Smartphones, insbesondere die Steuerung durch Gesten am Touchscreen hat unsere Gesellschaft geprägt. Während der Motion Capture den Bereich des Erfassens der Daten des Anwenders abdeckt, umfasst der Bereich von Human-Computer-Interaction (HCI) auch die Verarbeitung der erfassten Daten. Das Spektrum an Computer umfasst dabei vom Smartphone bis hin zu einem gewöhnlichen Rechner. Die einfachste Relation ist dabei ein Computer zu einem Menschen und die komplexeste Assoziation ist mehrere Menschen zu mehreren Computern. Die Interaktion kann durch verschiedene Steuerungsmöglichkeiten wie Tastatur und Maus oder auch durch Gestenerkennung gesteuert werden.

“Gesture recognition is important for developing an attractive alternative to prevalent human-computer interaction (HCI) modalities. Recognizing gestures is a complex task that involves many aspects such as motion modeling, motion analysis, pattern recognition and machine learning, even psycholinguistic studies.” - [Yu Yuan 2008]



Abbildung 6: BMW Cockpit

Quelle: [BMW 7er, 2015]

Die Abbildung zeigt, dass per Geste bestimmte Funktionen wie das Anrufen in dem neuen 7er BMW eingeleitet werden können.

2.4 Ubiquitous Computing

Ubiquitous Computing ist das Zeitalter, in dem ein Anwender mehrere Computer im Alltag benutzt. Diese Computer sind in unterschiedlichen Objekten implementiert und erledigen Funktionen des Menschen. Eingeleitet können die Interaktionen zwischen Mensch und Computer durch aktive Teilnahme oder auch ohne die Wahrnehmung des Menschen. Zu Beginn der Informationstechnik, in der ersten Phase, kam auf vielen Menschen nur ein Computer zu, da die Technik nicht erschwinglich war. Die zweite Phase ist das Zeitalter des Personal Computers (PC). Wie es aus dem Begriff zu entnehmen ist, hat der Menschen seinen persönlichen Computer. Die dritte Phase ist die Ära des Ubiquitous Computing, in der Anwender mit vielen Computern interagiert. Nur eine Teilmenge der Computer sind Personal Computer, ein Großteil der Computer machen eingebettete Rechner aus, die für spezifische Aufgaben bestimmt sind.

„The third wave of computing is that of ubiquitous computing, whose cross-over point with personal computing will be around 2005-2020. The „UC“ era will have lots of computers sharing each of us. Some of these computers will be the hundreds we may access in the course of a few minutes of Internet browsing. Others will be imbedded in walls, chairs, clothing, light switches, cars - in everything. UC is fundamentally characterized by the connection of things in the world with computation. This will take place at many scales, including the microscopic“ – [Weiser und Brown 1997]

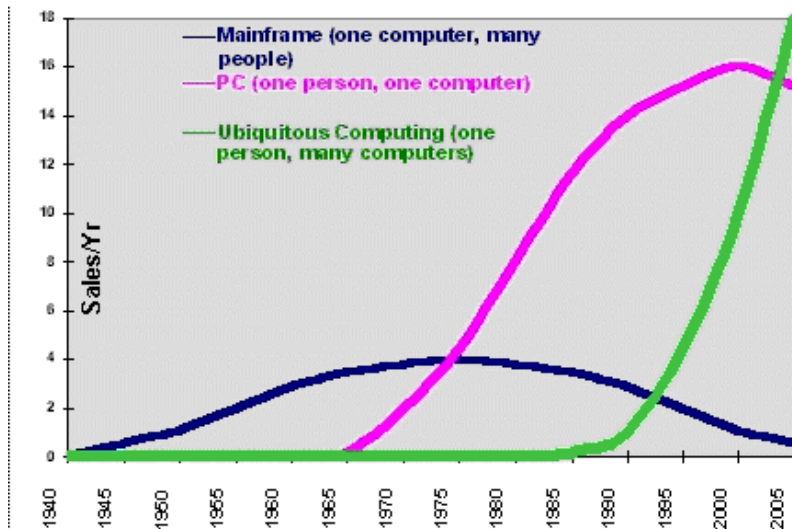


Abbildung 7: Verbreitung der Computer
 Quelle: [Inha University 2015]

Die Abbildung verbildlicht den stetigen Zuwachs der Computer, die auf einen Menschen kommen. Die grüne Linie, welche für die Ubiquitous Computing Phase steht, zeigt einen enormen Zuwachs mit der Zeit, während Computer, die für viele Anwender bestimmt sind, gegen Null verläuft. Die Anzahl der persönlichen Computer (PC) hat im Jahre 2000 den Höhepunkt erreicht.

2.5 Smart Home

Eine adäquate Übersetzung von Smart Home ins Deutsche ist das intelligente Haus. Ein primäres Ziel in dieser Umgebung ist die Lebensqualität als auch die Wohnqualität des Bewohners zu erhöhen. Das Konzept des Smart Home umfasst die verschiedenen Themengebiete der Sicherheit, Energieeffizienz, Umwelt, Komfort, Haushalt und Geräte. In dieser Arbeit geht es um die Steuerung dieser Geräte. Ein Smart Home verhält sich zu einem gewöhnlichen Haus, wie ein Smartphone zu einem gewöhnlichen Handy. Wie im Kapitel Ubiquitous Computing beschrieben, ist eine Vielzahl von verschiedenen Geräten in unterschiedlichen Bereichen vernetzt. Dies ermöglicht die Grundlage für die zentrale Steuerung dieser Komponenten. In der Praxis können Komponenten wie Musik-Anlagen, die über das Netzwerk zu erreichen sind, bequem gesteuert werden. Diese Objekte sind unter anderem beispielsweise das Auto, der Fernseher, die Jalousien oder die LED-Raumbeleuchtung. Ein Szenario, welches das Konzept der Smart-Home Umgebung wiedergibt, ist, dass der Anwender das zuvor geparkte Auto vom Wohnzimmer aus die Standheizung beispielsweise aktivieren kann.

„Das Smart Home ist ein privat genutztes Heim (z.B. Eigenheim, Mietwohnung), in dem die zahlreichen Geräte der Hausautomation (wie Heizung, Beleuchtung, Belüftung), Haushaltstechnik (wie z.B. Kühlschrank, Waschmaschine), Konsumelektronik und Kommunikationseinrichtungen zu intelligenten Gegenständen werden, die sich an den Bedürfnissen der Bewohner orientieren. Durch Vernetzung dieser Gegenstände untereinander können neue Assistenzfunktionen und Dienste zum Nutzen des Bewohners bereitgestellt werden und einen Mehrwert generieren, der über den einzelnen Nutzen der im Haus vorhandenen Anwendungen hinausgeht“ –[Strese et al. 2010]

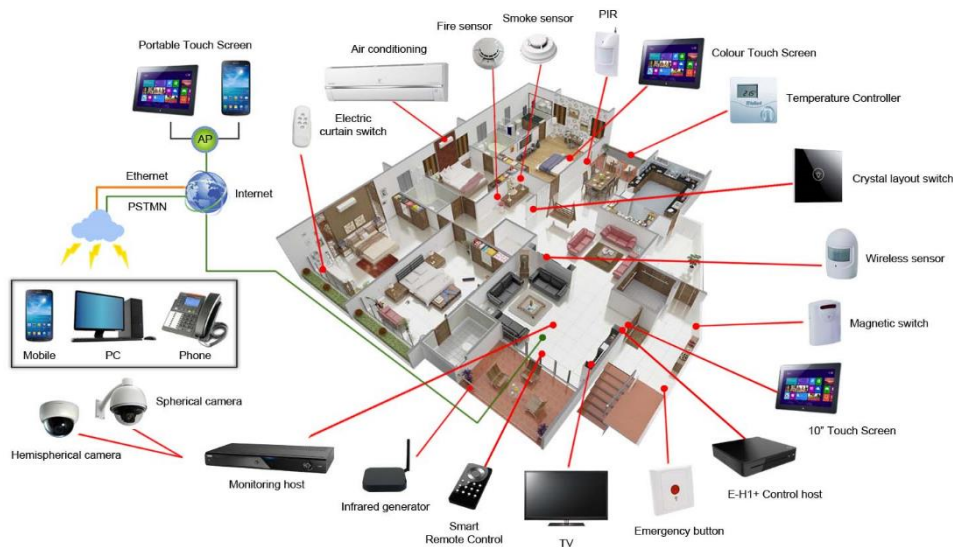


Abbildung 8: Mögliche Smart Home Komponenten

Quelle: [Witura.com Smart Home]

Die Abbildung visualisiert die wichtigen Komponenten der Smart Home Umgebung, die zum Einsatz kommen. Dabei umfassen die Geräte die Aspekte Sicherheit, Entertainment, Haushalt und Kommunikation. Als Eingabegeräte für die Steuerung der Komponenten sind Fernbedienung (Controller), Smartphone und Tablet in der Abbildung aufgezeigt. Die Alternative zu den aufgeführten Steuerungseinheiten ist die Gestenerkennung, die in dieser Arbeit thematisiert wird. Der Clou an der Gestensteuerung ist, dass keine unpraktischen Hardwaregeräte wie Fernbedienung mit transportiert werden, was die Steuerung von Gesten räumlich gesehen nahtlos und somit harmonisch macht.

3. Analyse

In der Realität gibt es keine konventionellen Gesten, die optisch via Kamerasensoren erfasst werden, um bestimmte Funktionen eines Geräts anzusprechen, wie beispielsweise bei Touchscreens die Funktion "*pinch to zoom*" gegeben ist. Aus dieser Grundlage heraus ist es wichtig, dass das Repertoire an Gesten möglichst Smart Home konform zu konzipieren ist. Das bedeutet, dass die Gesten die konzeptionellen Ideen des Smart Homes wie Bequemlichkeit, Benutzerfreundlichkeit und Einfachheit nicht verstoßen darf. Um das zu gewährleisten, wurden Kriterien definiert, die unten aufgeführt sind.

3.1 Konzept zur Steuerung

Ein Konzept der Smart Home Umgebung ist, dass unterschiedliche Geräte wie LED-Beleuchtung, Musik-Anlage und andere Geräte schnell angesprochen werden können, um das gewünschte Ergebnis zu erzielen. Am Anfang muss eine Auswahl getroffen werden, um die Steuerung dieser Komponente vorzunehmen. Eine Auswahl wird getroffen, wenn die definierte Geste des Gestenalphabetes durchgeführt wird. Ist die Auswahl der zu steuernden Komponente erfolgt, wird die Komponente reguliert. Die Komponenten können Unterschiede in der Regulierung aufweisen. Ein konkretes Beispiel für unterschiedliche Regulierungen zwischen zwei Komponenten ist die Musik-Anlage in Relation zu einer LED-Beleuchtung. Bei der Steuerung der Musikanlage sind vier Befehle für die Steuerung essentiell, nämlich das Wechseln der Titel bzw. der Alben, die Regulierung der Lautstärke, das Abspielen und Stoppen der Musik selbst und das Auswählen (bestätigen). Bei LED-Beleuchtung ist hingegen die Regulierung der Farbe durch die RGB Farbwerte, die Sättigung und die Helligkeit essentiell.

3.2 Eingabealphabet in 2D

Die Entwicklung der berührungsempfindlichen Bildschirme (Touchscreen) lösten die mechanischen Tasten von vielen Computern ab. Die ersten nennenswerten Computer, die es in den Alltag des Menschen geschafft haben und ohne Eingabetasten auskommen, sind die Personal Digital Assistant (PDA). Die kompakten Computer konnten die Handschrift der Benutzer, die mittels Stift eingegeben wurde, interpretieren. Der Platz für die Tastaturen konnte somit für den Bildschirm ausgenutzt werden. Später ersetzten die Handys diese Geräte.



Abbildung 9: PDA Messagepad 2000
[Wikimedia.org Apple Newton MessagePad, 2000]

3.3 Unistroke

Unistroke ist ein 2D Eingabe-Alphabet für Gesten. Erstmals wurde Unistroke 1993 von Goldberg und Richardson (Goldberg u. Richardson 1993) zur Benutzung für Lightpens vorgeschlagen. Das Alphabet besteht aus durchgezogenen Linien, die jeweils für einen Buchstaben stehen. Der Kreis an den Buchstaben markiert den Anfang. Unistroke ist längst in der Wissenschaft präsent und wird unter anderem für HCI verwendet.

Unter der Annahme, dass die Gesten förmlich in die Luft gezeichnet werden und Unistroke als Eingabealphabet für Gestenerkennung bestimmt ist, müssen Kontrollmechanismen eingeführt werden um die Fehlinterpretation zu vermeiden. Das hintereinander Ausführen der Buchstaben kann ansonsten zu Fehlinterpretation führen, wenn die Start-Stopp-Problematik (siehe folgenden Abschnitt) nicht geregelt ist. Folgen beispielsweise die Buchstaben „A“ und „T“ hintereinander, kann das System die Zeichenfolge „AT“ interpretieren oder den Buchstaben „T“. Ist der Kontrollmechanismus so geregelt, dass die Start-Stopp-Problematik gelöst ist, kann die Entscheidung besser getroffen werden. In der Praxis markiert die Dauer der Betätigung der Tasten den Anfang und das Ende einer Geste bzw. eines Buchstaben.

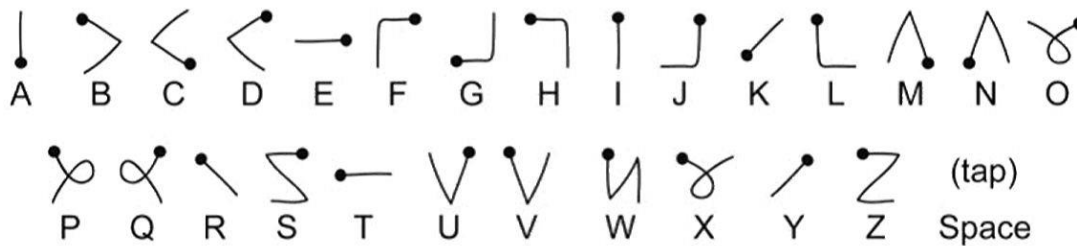


Abbildung 10: Unistroke
 Quelle: [EECS Unistroke, 2014]

3.4 Start-Stopp-Problematik

Wie aus dem Kapitel 2.1 hervorgeht, sind Gesten streng genommen alle Handbewegungen des Anwenders. Eine definierte Abfolge von Bewegungen ist in diesem Kontext die Geste. In dieser Arbeit sind die Gesten im Kontext der Gestenerkennung zu verstehen. Eine Geste hat einen Anfang und ein Ende, auch wenn die Hand weiterhin in Bewegung ist. Bei Touchscreen ist der Anfang und das Ende durch das Berühren definiert. Gesten, die förmlich in die Luft gezeichnet werden, haben zunächst keinen Anfang und kein Ende, was die Erkennung schwieriger macht. Den Anfang und das Ende einer Geste können verschiedene Dinge markieren. Beispielweise kann die Zeitdauer der gedrückten Taste an einem Controller, wie es an der Wii Spielkonsole gegeben ist, den Beginn und das Ende der Geste deklarieren. Das gleiche Prinzip gilt auch für bestimmte Lightpens, die das Einschalten und Ausschalten des Leuchtmittels benutzen, um den Anfang und das Ende zu markieren. Der Nachteil von solchen Lösungen ist, dass Hardware wie ein Stift und ein Controller getragen werden muss, was unpraktisch ist und deshalb in das Konzept des Smart Homes nicht passt. Mit der Kinect ist es möglich, Unterscheidungen zu treffen, ob die Handfläche des Anwenders zu sehen ist oder der Anwender förmlich nach etwas greift. Die Finger sind dabei geballt, wie bei einer Faust, und dieser Modus nennt sich in Kinect „grab“. Diese Unterscheidung kann benutzt werden, um Gesten einzuleiten. M.a.W. kann die Zeitdauer der Faust auch die Zeitdauer der Geste bedeuten. Außerdem ist es mit dem Tiefensensor von Kinect möglich, die Entfernung vom Sensor zum Anwender zu bestimmen. Zudem kann auch die Entfernung in der Tiefe von der Hand zum Oberkörper im dreidimensionalen Raum bestimmt werden. Unter der Annahme, dass ein bestimmter Schwellwert die Eingabezone markiert, kann das Überschreiten und die Unterschreitung der Hand dieser Eingabezone, die Gestendauer definieren. Die Handbewegung des Anwenders ist eine Folge von 2D-Koordinaten, die in bestimmten Intervallen folgen. Anhand der durchgeführten Geste des Anwenders und der Geste aus dem Gestenalphabet kann die Abweichung errechnet werden. Die Abweichung kann man als ein float Wert zurückgeben. Meist entspricht der Wert 1.0, eine hundertprozentige Übereinstimmung, was in der Realität sehr unwahrscheinlich ist. Eine Erkennung, die etwa bei 85% (0.85) liegt, ist durchschnittlich.



Abbildung 11: Wii Remote Controller
Quelle: [Wikipedia Wii remote, 2014]

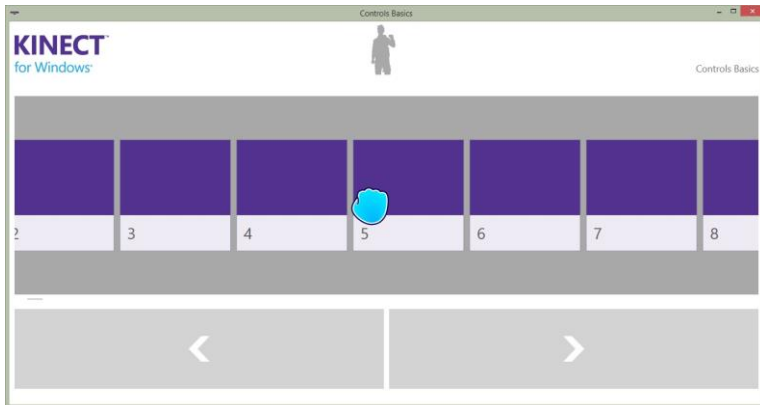


Abbildung 12: Kinect Developer Toolkit 1.8 Sample Controls Grap
Softwarequelle: [Microsoft Kinect Developer Toolkit 1.8]

Die Abbildung gibt die Hand des Anwenders wieder. Die Hände „greifen“ förmlich das Objekt. Die Finger sind zusammengeballt wie bei einer Faust. Das Toolkit steht auf der Homepage von Microsoft zur Verfügung.

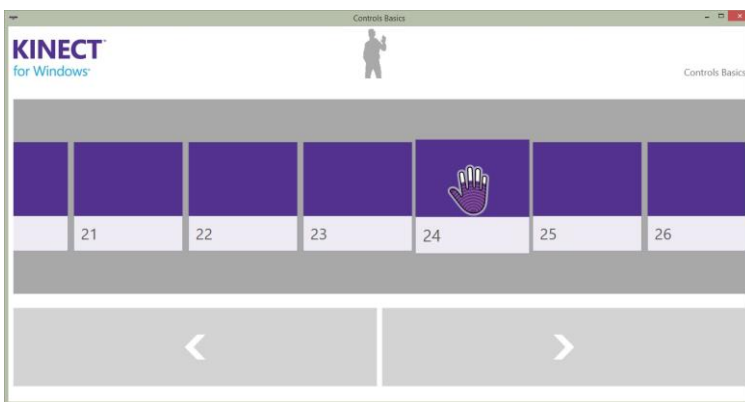


Abbildung 13: Kinect Developer Toolkit 1.8 Sample Controls Distance

Die Abbildung gibt die Distanz zwischen der Hand und dem Oberkörper des Anwenders wieder. Ist die Hand weit genug ausgestreckt, kann das Objekt ausgewählt werden.

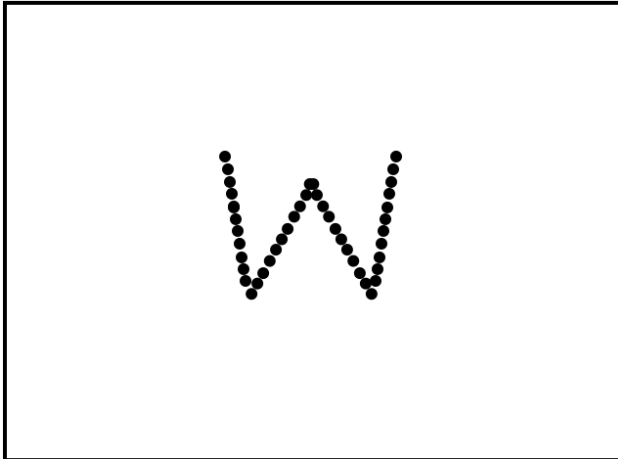


Abbildung 14: Die definierte Geste „W“

Die Abbildung gibt die definierte Geste „W“ wieder. Eine Geste wird meist durch eine Aufnahmefunktion definiert. Alle definierten Gesten zusammen ergeben das Gestenalphabet. Die Aufnahmefunktion speichert die Positionen der Hand vom Anwender, die später für die Erkennung unverzichtbar ist. Zeitlich entspricht ein Kreis in der Abbildung ein Frame der Aufnahme. Die Positionen wurden geglättet, um später eine genauere Erkennung zu ermöglichen.

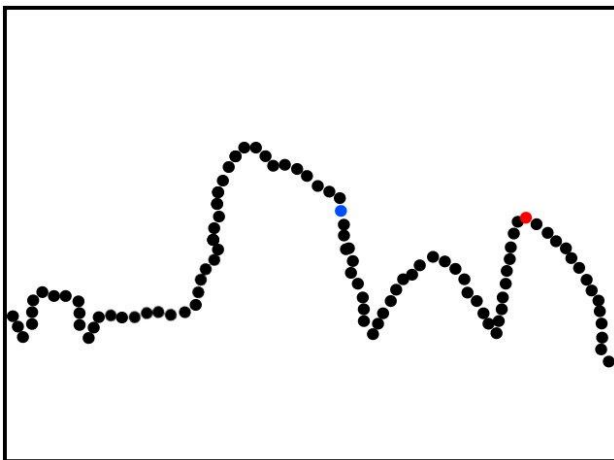


Abbildung 15: Die ausgeführte Geste „W“

Die Abbildung gibt eine Sequenz der Handbewegungen vom Anwender wieder. Die Positionen sind natürlicher, da das Ergebnis nicht geglättet ist. Der blaue Kreis ist der Anfang von der definierten Geste und der rote Kreis entsprechend das Ende. Der Intellekt des Menschen ist in der Lage die definierte Geste „W“ aus der tatsächlichen Bewegung zu erkennen. Ist die Start-Stopp Problematik nicht gelöst, müssen alle Punkte gewertet werden, um den Anfang und das Ende der Geste zu bestimmen, was mit sehr viel Rechenaufwand verbunden ist. Unter der Annahme, dass die Geste „W“ ausgeführt wird, wenn die Hand geballt ist (Modus grab), dann hat die Geste einen Anfang und ein Ende und entsprechend können Gesten verglichen werden.

3.5 \$1 unistroke recognize

Unter dem Namen „Gestures without Libraries, Toolkits or Training: A \$1 Recognizer for User Interface Prototypes“ gibt es eine wissenschaftliche Ausarbeitung von Jacob O. Wobbrock, Andrew D. Wilson und Yang Li (Wobbrock u.a. 2015) auf der Webseite von der von University of Washington, die prinzipiell wie diese Arbeit ist. Die praktische Ausarbeitung „\$1 unistroke Recognizer“ ist online. Das Tool verwendet unter anderem den Dynamic-Time-Warping Algorithmus. Die Gesten werden durch die Maus definiert. Durch das Klicken der linken Maustaste wird die Geste eingeleitet.



Abbildung 16: \$1 unistroke recognize DTW Record
Die Geste „M“ wird definiert.

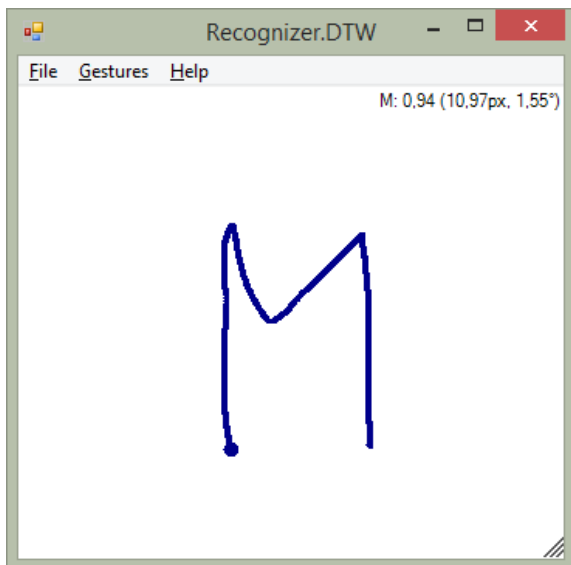


Abbildung 17: \$1 unistroke recognize DTW Recognize
Die ausgeführte Geste M ist zu 94% identisch zur definierten Geste „M“.

3.6 Konzeptionelle Gestenanforderungen

Es wurden Anforderungen definiert, um die Erkennung der Gesten zu gewährleisten, als auch praktisch zu halten. Bei der Definition der Gesten wurden die Aspekte Benutzerfreundlichkeit und programmiertechnische Umsetzung berücksichtigt.

Folgende Kriterien wurden aufgestellt:

1. Wohl unterscheidbar
2. Intuitiv
3. Erweiterbar
4. Performance

1. Wohl unterscheidbar

Das erste Kriterium ist aus programmiertechnischen Gründen deklariert worden, denn unter der Annahme, dass Gesten in der Bewegung sich stark ähneln, ist die Wahrscheinlichkeit der fehlerhaften Interpretation der Geste höher. Mit anderen Worten müssen die Gesten in der Bewegung starke Unterschiede aufweisen. Gesten hingegen, die grundlegende Unterschiede aufweisen, sind einfacher zu interpretieren, was die fehlerhafte Interpretation verringert. Ein konkretes Beispiel, welches das Kriterium wohl unterscheidbar nicht erfüllt, sind die beiden Gesten, die die Bewegungen "1" und "I" abbilden. Der Unterschied zwischen den beiden Gesten ist minimal. Alle Gesten, die im System deklariert werden, müssen wohl unterscheidbar sein. Zusätzlich darf die Geste keiner Bewegung des Alltags entsprechen oder ähneln. Mit natürlichen Bewegungen sind Tätigkeiten wie das Telefonieren, die Bewegung der Hände beim Gehen, beim Zappen, Jubeln und dergleichen gemeint.

2. Intuitiv

Das Ziel ist ein Gestenkonzept zu generieren, welches "handbuchfrei" ist und nicht der Erklärung bedarf. Das Ziel ist, dass wenn der Anwender das Konzept „Ein Buchstabe, eine Komponente“ verstanden hat, ohne die zu Hilfenahme eines Handbuchs alle Komponente eigenständig auswählen und regulieren kann. Die intuitive Bedienung des Systems ist eine wichtige Eigenschaft des Smart Homes, denn die Intuition erhöht die Benutzerfreundlichkeit.

3. Erweiterbar

Vorgesehen ist, dass die gängigen Geräte wie der Fernseher, die Beleuchtung, die Musik-Anlage und dergleichen schon mit vordefinierten Gesten ansprechbar sind. Neue Geräte, die an das System angeschlossen werden, müssen mit neuen Gesten verknüpft werden. Unter der Annahme, dass zehn verschiedene Gerätetypen an das System angeschlossen werden, müssen auch exakt zehn Gesten deklariert werden.

4. Performance

Die Gesten des Anwenders müssen schnell erkannt werden und dies gilt auch für die Regulierung der Einstellungen. Um das zu gewährleisten, muss der Algorithmus schnell erfolgen. Eine Gestensteuerung, die langsamer ist als die herkömmliche Steuerung, ist nicht von Vorteil, da der Faktor Zeit eine sehr große Rolle in Smart Home Umgebung spielt.

3.6.1 Konzept „Ein Buchstabe eine Komponente“

Grundsätzlich sind Objekte, die in einem Haus elektrisch gesteuert werden können, potenzielle Komponenten des Systems. Gängige Geräte, die bereits in heutigen Häusern gesteuert werden, sind Beleuchtungen, Fenster, Musikanlagen, Fernseher und Jalousien. Einige Objekte können mehrmals in einem Haushalt vorkommen. Jede konventionelle Wohnung hat mehrere Fenster, diese werden aber als eine Komponente des Systems betrachtet. Alle Komponenten des Systems werden jeweils exakt mit einer definierten Geste direkt ausgewählt. Ziel des Konzepts ist es primär jede Komponente mit der definierten Geste, die dem Anfangsbuchstaben entspricht oder ähnelt, zu wählen.

3.6.2 Auswählen der Komponente

In einer Smart-Home Umgebung kommt der Anwender mit unterschiedlichsten Geräten in Berührung. In einem normalen Umfeld, das nicht einer Smart-Home Umgebung entspricht, wird die Steuerung unterschiedlicher Geräte durch verschiedene Eingabegeräte bewerkstelligt. In der Realität ist meistens genau ein Schalter, ein Knopf oder eine Fernbedienung für die Steuerung eines bestimmten Geräts ausgerichtet. Werden die Fernbedienungen bzw. die Knöpfe in einer Wohnung durch eine definierte Geste ersetzt, können die Geräte gesteuert werden, auch wenn die Hardware wie Fernbedienung nicht griffbereit ist. Praktisch bedeutet das, dass der Anwender durch die vordefinierte Geste genau ein Gerät auswählen kann.

Ausgehend von der Idee "Ein Buchstabe, eine Komponente" ist das Konzept, dass der Anwender beispielsweise das Licht passend zu seiner Stimmung einstellen kann, in dem die Geste "L" für Light förmlich in die Luft malt. Entsprechend ist die Geste "T" für TV und "J" für die Jalousien konzipiert. Wird das System durch eine Komponente erweitert, dessen Anfangsbuchstabe bereits verwendet wird, wie es bei "Gallery" und "garage door" der Fall ist, muss der Anwender auf ein anderes Wort wie "door" ausweichen, damit jede Komponente durch einen Buchstaben ausgewählt werden kann. Im Sonderfall können Gesten zum Auswählen der Komponente von dem Buchstabenkonzept ausweichen, wenn beispielsweise keine plausiblen Buchstaben zu vergeben sind.

Neben der Möglichkeit direkt durch die entsprechende Geste des Anfangsbuchstaben die Komponente zu steuern, gibt es die zweite Möglichkeit, durch die vordefinierte Geste "O", was für Overview steht, die Übersicht aller Geräte auszugeben, die angesteuert werden können. Die zweite Möglichkeit ist gerade für Anfänger des Systems hilfreich, da die Geräte mit der entsprechenden Geste aufgezeigt werden, die für die Steuerung notwendig ist. Zudem kann der Anwender durch das Bewegen der Hand in Richtung der beabsichtigten Komponente, das gewünschte Gerät auswählen.



Abbildung 18: Geste "O" öffnet das Menü Overview.



Abbildung 19: Overview

In der Abbildung ist das Auswahlmü (Overview) zu sehen. Ausgewählt wurde Light. Anschließend kann das Licht eingestellt werden.



Abbildung 20: Gesture „L“

Die Geste "L" steht für Light. Diese Möglichkeit ist der direkte Weg im Vergleich zu Auswahlmenü.

Mit anderen Worten können Komponenten durch zwei Wege angesteuert werden. Die Erste ist die Geste "O", was die Overview Ansicht erzeugt. Anschließend kann der Anwender durch das Bewegen der Hand in Richtung der Komponente entsprechend auswählen. Der zweite und der direkte Weg ist die Geste der entsprechenden Komponente wie z.B. "L" für die Lichtsteuerung oder "M" für die Steuerung der Musikanlage.

3.6.3 Regulierung der Parameter

Wie im Kapitel 3.1 beschrieben, sind verschiedene Parameterwerte für unterschiedliche Geräte zu übergeben. Unter Parameter sind Werte (Befehle) für das Einstellen der Lautstärke, des Farbtons, der Helligkeit, Programm weiter, Programm zurück, Titel weiter, Titel zurück, Jalousien hoch oder Jalousien runter und dergleichen zu verstehen. In diesem Abschnitt wird sowohl die Regulierung der LED Beleuchtung, als auch die Steuerung der Musik-Anlage als Beispiel beschrieben.

Ist die Auswahl durch die Geste "L" getroffen oder durch das Auswählen im Menü über die Übersicht "O" (Overview) das Licht ausgewählt, können die Lichteinstellungen vorgenommen werden. Ein passendes Farbmodell für die Lichteinstellung ist das Farbmodell HSV (Hue Saturation Value). Die Werte für Farbe, Sättigung, und Helligkeit können einzeln abgestimmt werden. An anderes Farbmodell, das speziell bei LED-Beleuchtungen auch gängig ist, ist der RGB Farbraum. In diesem Farbraum ist der Nachteil, dass die Helligkeit nicht explizit berücksichtigt wird. In dem praktischen Teil wird das RGB Farbmodell für die Abstimmung des Lichts verwendet.

Im RGB Farbraum sind 0-255 Werte für jeweils rot (R), grün (G), blau (B) vorgesehen. Damit sind $256^3 = 16.777.216$ Farbeinstellungen theoretisch möglich. Werden die RGB-Werte einzeln reguliert, so muss dreimal eine Auswahl getroffen werden, was zeitaufwendig ist und es besteht die Möglichkeit, dass der gewünschte Farbton nicht getroffen wird. Ist der Farbraum vertikal und linear geordnet, so kann die Auswahl schneller erfolgen, da die Übersicht der möglichen, unterscheidbaren Farben angezeigt wird und nur eine Auswahl getroffen werden muss. Mit anderen Worten gibt die Höhe der Y-Achse den Farbwert aus RGB wieder.

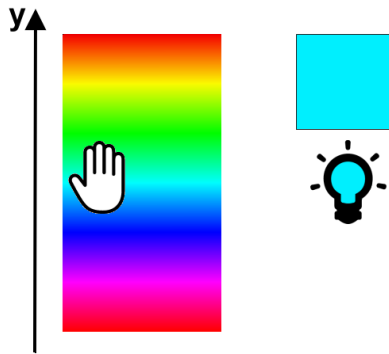


Abbildung 21: Vertikales Farbstreifen

Bei der Regelung der Lautstärke für die Musikanlage ist eine Skala von 0-100 vorgesehen. Bei so einer großen Auswahlmöglichkeit ist es wichtig, dass die Geste schnell zur anvisierten Lautstärke führt. Die Geste für die Regulierung der Lautstärke ist die horizontale Bewegung der Handfläche, was funktional betrachtet einem Lautstärkeregler entspricht. Eine alternative Geste ist die rotierende Bewegung der Handfläche. Im praktischen Teil der Arbeit wurde die erste Variante umgesetzt. Beide Gesten passen jedoch zum Konzept für die Gestenerkennung dieser Arbeit. Es handelt sich hier um ein Beispiel, wie die Regulierung ablaufen könnte.



Abbildung 22: Lautstärke einstellen

Bei der Abbildung 11 ist das klassische rechtwinklige Dreieck für das Lautstärkesymbol zu sehen. Das Bewegen der Hand in die rechte Richtung (x-Achse) erhöht die Lautstärke. Die Möglichkeit einen Musiktitel bzw. ein Album zu wählen, ändert sich je nach Aufbaumechanismus. In der Praxis sind zwei Möglichkeiten bei Musikanlagen vorzufinden.

Die erste Möglichkeit ist die Anordnung einer vertikalen Liste der Titel bzw. der Alben. Passende Geste, um einen Titel aus der Liste zu markieren, wäre die vertikale Bewegung von der Hand. Verbildlicht bedeutet das, dass die Hand auf der Höhe der Titel ist. Hält man die Hand auf der gleichen z.B. einige cm über dem Kopf für ein paar Sekunden, könnte das die Bedingung erfüllen, die Liste nach oben zu scrollen. Diese Möglichkeit erfordert die Funktion, den markierten Titel auszuwählen.

Die zweite Möglichkeit ist das einfache Switchen der Titel, was von der funktionsweise ähnlich wie beim Fernseher zu zappen ist. Der Anwender kann durch den Gestenbefehl einen Titel weiter oder zurück gelangen. Angebrachte Geste dafür ist die schnelle Handbewegung von der Mitte des Körpers nach außen. Je nach Definition könnte die Bewegung von der Mitte des Körpers nach links, z.B. "nächster Titel" bedeuten und umgekehrt. Dabei ist die schnelle Bewegung nach außen relevant für das Zappen bzw. Wechseln der Titel. Wird diese Bewegung zu langsam durchgeführt, wird kein Befehl ausgeführt.

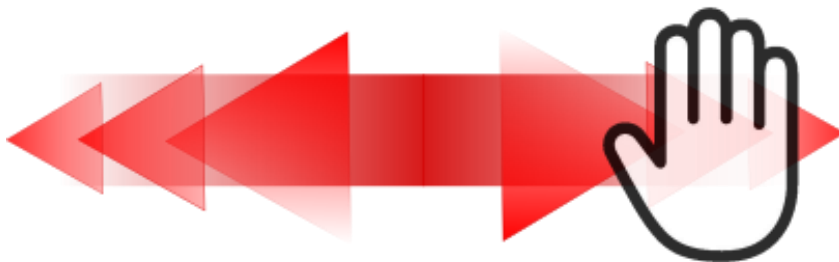


Abbildung 23: Geste das Wechseln der Titel oder der Programme

3.6.4 Das Auswählen

Bei der Regulierung der Lautstärke und dem Wechseln der Titel bzw. Alben aus der Liste, ist ein Auswahlmechanismus durch eine Geste zwingend notwendig, da die Hand die gewünschten Parameter erst einstellt, wenn die Auswahlgeste erfolgt ist. Passende Auswahlmechanismen sind das Zuklappen der Finger zu einer Faust oder das Berühren/Verlassen der Parameter selbst (siehe Abbildung).

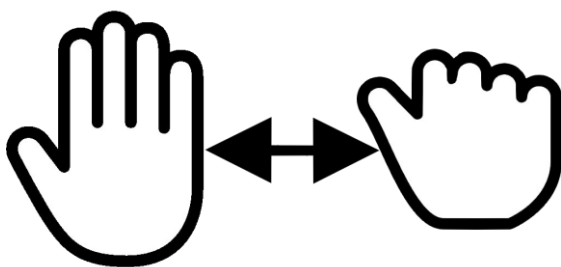


Abbildung 24: Geste für das Auswählen

Abspielen und Stoppen

Um die Funktionen Abspielen und Stoppen zu ermöglichen, reicht eine Geste, die definiert werden muss.

Spielt die Anlage die Musik schon ab, kann mit der vordefinierten Geste die Music gestoppt werden. Erfolgt anschließend der gleiche Befehl wieder, wird die Musik weitergespielt. Die Alternative ist jeweils für das Abspielen und für das Stoppen verschiedene Gesten zu definieren.

3.7 Szenarien

Die unten folgenden Szenarien beschreiben die Möglichkeiten des Anwenders des gestenbasierten Systems. Es werden nur ein paar prägnante Beispiele aufzeigt, die nur ein Bruchteil der Möglichkeiten sind. Anhand der Beispiele wird das Konzept und die Vorteile verdeutlicht.

Fall 1: Das Garagentor

Der Anwender kommt nach einem langen Arbeitstag nach Hause und parkt das Auto in der Garage. Zu Hause, im Wohnzimmer auf der Couch macht er es sich gemütlich. Dem Anwender fällt ein, dass das Garagentor noch nicht geschlossen ist. Es wäre zu unpraktisch durch das ganze Haus zu gehen, nur um das Tor zu schließen. Abhilfe schafft ihm seine Smart Home Umgebung mit dem Konzept „Ein Buchstabe, eine Komponente“, denn er braucht nicht dorthin zu gehen. Der Anwender kann einfach auf der Couch das Garagentor schließen, indem er durch eine von ihm vordefinierte Geste es ansteuert und es schließt.

Fall 2: Der Fernseher

Auf der Couch möchte der Anwender den Fernseher einschalten, doch leider ist die Fernbedienung gefühlt viel zu weit weg und das Smartphone mit Infrarot (bzw. App) nicht in greifbarer Nähe. In dieser Situation kann der Anwender durch zeichnen des Buchstaben „T“ in die Luft, was für TV steht, den Fernseher einschalten und anschließend zappen.

Fall 3: Die Beleuchtung

Allmählich wird es dunkel und außerdem möchte der Anwender ein passendes Ambiente schaffen. Die Farbe Blau entspannt den Anwender. Was er dafür tun muss ist einfach die Geste „L“ auszuführen. Nun kann er die gewünschte Farbe der LED-Beleuchtung auswählen.

Fall 4: Die Musikanlage

Der Anwender erwartet Besuch und es muss gekocht werden. Beim Kochen hört der Anwender hin und wieder auch Radio. Mit Händen, die zu nass oder mit der Soße bekleckert sind, möchte der Anwender nicht die Musik-Anlage abspielen. Die Geste „M“, was die Musikanlage einschaltet, erspart dem Anwender das Waschen und das Trocknen der Hände.

Fall 5: Die Jalousien

Die Gäste sind eingetroffen und der Anwender bemerkt, dass die Jalousien nicht passend eingestellt sind. Der Anwender kann dank des integrierten Gestenerkennungssystems durch die Geste „J“ die gewünschte Regulierung vornehmen.

Fall 6: Die Besucher

Der Gast begibt sich ins Gästezimmer und möchte das Licht anpassen. Das Konzept „Ein Buchstabe, eine Komponente“ ist einfach und gilt auch für das Zimmer. Der Gast hat längst begriffen, dass die Geste „L“ für Light steht, und steuert im Nu das Licht.

4. Umgebung

In diesem Kapitel wird auf die Umgebung eingegangen. Die Hardwarekomponenten als auch die Software bzw. Programme, die zum Einsatz kommen, werden thematisiert. Es werden zudem die Einsatzgebiete aufgezeigt.

4.1. Einsatzgebiete

Die Einsatzgebiete des Systems sind primär Wohnungen als auch Häuser für den privaten Gebrauch. Das System ist für eine Smart Home Umgebung ausgelegt. Praktisch sind Räume, in denen die Erfassung der Bewegungen vom Anwender möglich sind, die Einsatzgebiete. Primär sind die Zimmer wie Wohnzimmer, Flur, Küche, Kinderzimmer vorgesehen. Je nach dem, was für ein Gerät bzw. Sensor die Bewegung erfasst, müssen Faktoren wie Licht und dergleichen beachtet werden, da ansonsten die Bewegungen nicht auswertbar sind.

„Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.“- [Acm Sigchi Curricula, 1992]

4.2 Hardware

In diesem Abschnitt werden die eingesetzten Hardwarekomponenten aufgezeigt. Man kann diese Geräte in die drei Bereiche unter Eingabe, Verarbeitung und Ausgabe (EVA-Prinzip) zuordnen. Denkbar sind auch andere Geräte, die die gleiche Funktion erfüllen.

4.2.1 Kinect Xbox 360

Der Kinect X-Box 360 ist eine Steuerungseinheit, die mit der Spielekonsole X-Box verbunden wird. Das Produkt wurde von der Firma Microsoft und PrimeSense entwickelt und wurde zum ersten Mal November 2010 verkauft. Kinect ermöglicht dem Anwender der Spielekonsole, mittels Körperbewegung das Spiel zu steuern. Alternativ kann die Hardware auch an einem Computer angeschlossen werden, um gesteuert zu werden, wie es in dieser Arbeit der Fall ist.

1. 3D-Tiefensensor.
2. RGB Kamera
3. Mikrofone-Array
4. Motorisierte Neigung

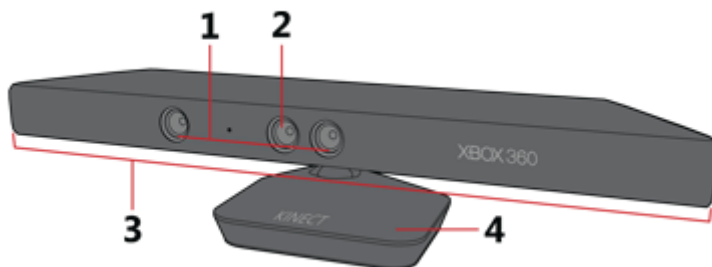


Abbildung 25: X-Box Kinect 360

Quelle: [Xbox Kinect 360]

3D Tiefensensor

Der Kinect beinhaltet ein 3D Tiefensensor, der mit Infrarot arbeitet. Dieser Sensor sendet Infrarot Strahlen, die später durch die Reflektion erfasst werden. Anschließend kann je nach Intensität des Strahls die Entfernung bestimmt werden. Die Daten werden zu einem Depth-Stream verarbeitet.

Farbkamera

Zudem besitzt die Hardware zwei RGB-Kameras, die zur Bildaufnahme dienen. Dieser Sensor arbeitet mit RGB, wobei R für red, G für green, B für blue steht. Die Auflösung der Kameras beträgt 640x480 Pixel.

Mikrofon-Array

Ein Mikrofon-Array aus 4 Mikrofonen mit einer Abtastrate von 16kHz ist verbaut. Die Richtung der Töne kann durch die vier Mikrofone ermittelt werden. Die Mikrofone sind nicht relevant für die Arbeit.

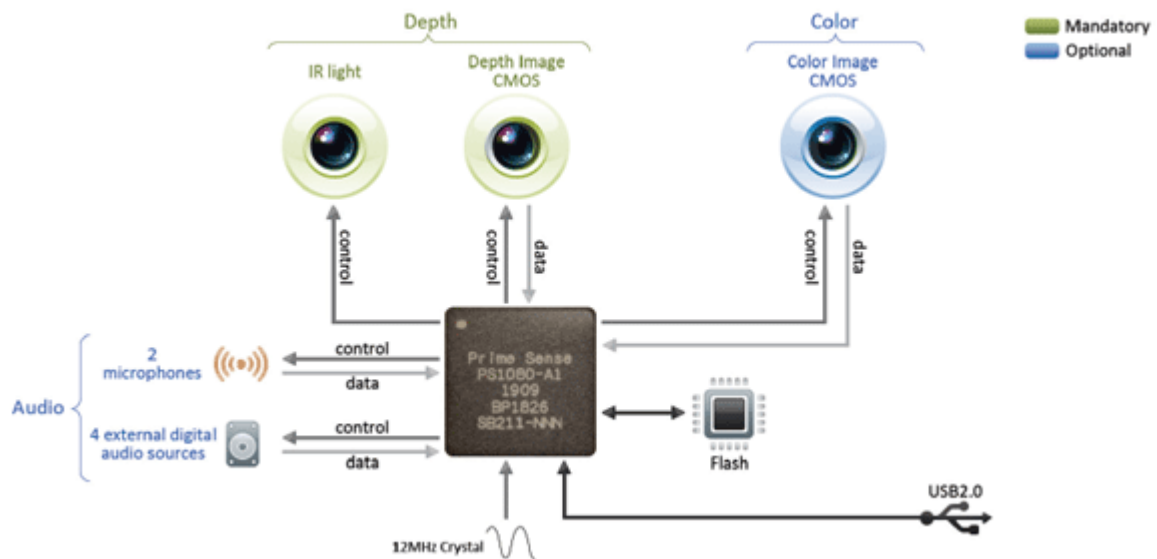


Abbildung 26: X-Box Kinect Sensors
 Quelle: [PrimeSense Ltd, 2010)]



Abbildung 27: X-Box 360 Kinect
 Quelle: [Xbox Kinect 360 Support]

4.2.2 Arduino Due

Das Arduino Due Board ist die Schnittstelle zwischen dem Computer und der LED Beleuchtung. Das Board kann über die sogenannten Ausgangs- und Eingangspins Daten empfangen und senden. Das Eingabegerät Kinect Xbox 360 nimmt die Bewegung auf und sendet je nach Geste die Informationen über die serielle Schnittstelle an den Controller. Die Informationen, die durch den Eingangspin empfangen werden, sind in dem Fall die RGB Werte. Diese Daten werden dann für das Ausgabegerät entsprechend bearbeitet und über den Ausgangspin versendet. Es können auch andere Controller von anderen Firmen wie Raspberry

Pi und dergleichen benutzt werden, soweit die technischen Voraussetzungen gegeben sind. Es muss sichergestellt werden, dass die Treiber installiert sind. Der Controller wird per USB-Kabel mit dem Computer verbunden und für die Programmierung wird die IDE von Arduino benötigt.

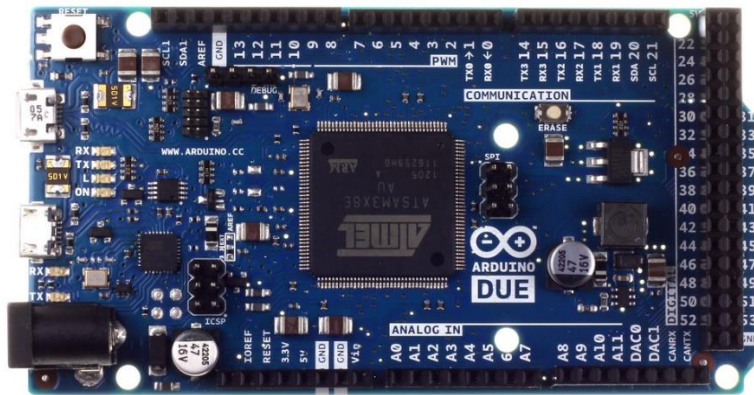


Abbildung 28: Arduino Due
Quelle: [Pololu.com Arduino]

4.2.3 LED Strip (WS2812B)

Die LED Strip (Beleuchtung) ist der Empfänger des Bitstroms und ist auch gleichzeitig das Ausgangsgerät. Die LED-Beleuchtung mit 60 LED gibt die ausgewählten Farben des Anwenders wieder. Es handelt sich hierbei um RGB Werte. Bei Bedarf können die Farben des einzelnen LEDs einzeln angesprochen werden. Damit ist die LED Strip auch Ambientlicht fähig. Für den Zweck der Arbeit werden alle LEDs auf die gleiche Farbe gesetzt. Zu beachten ist, dass die Anschlüsse richtig verdrahtet bzw. gelötet werden.

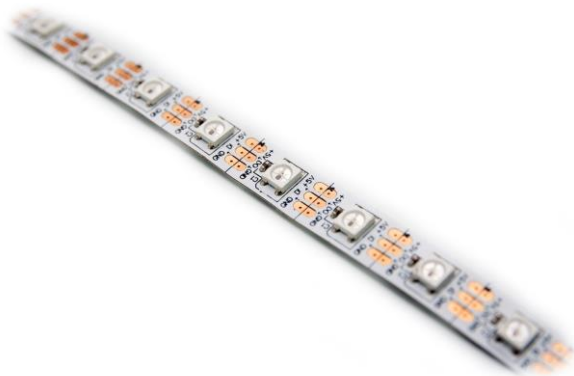


Abbildung 29: LED Strip (WS2812B)
Quelle: [FPV1]

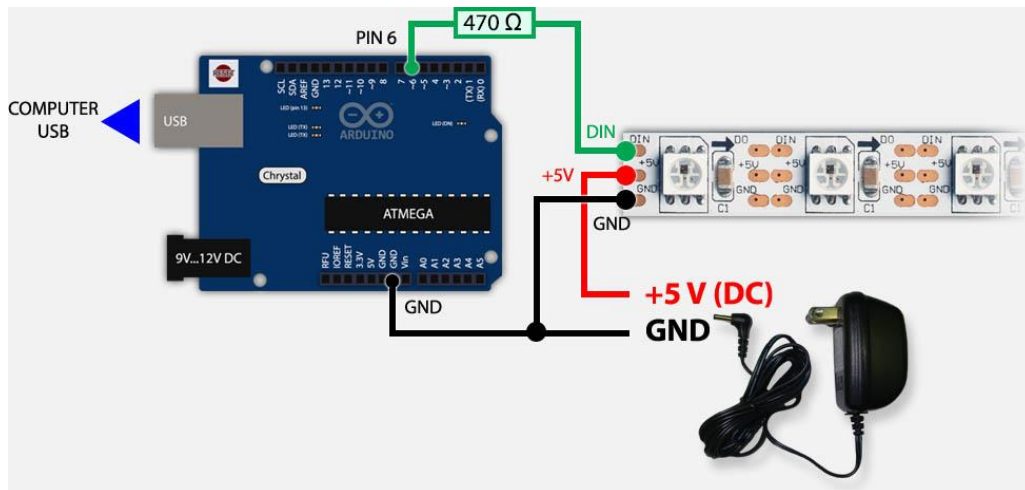


Abbildung 30: Aufbau Arduino und LED Strip

Quelle: [Teawiking4all 2014]

Für diese Arbeit wurde die Steuerung der LED-Beleuchtung als Komponente ausgewählt und entwickelt, um die Funktionsweise des Systems aufzuzeigen. Das System kann mit anderen Komponenten erweitert werden. Dazu müssen die Komponenten digital steuerbar sein.

Der Anwender des Systems befindet sich in Smart Home Umgebung, in einem Zimmer und steht gegenüber vom Kinect-Sensor, welches die Bewegungen aufnimmt. Die Bewegungen vom Anwender werden vom Kinect erfasst und an den Computer über die USB-Schnittstelle gesendet. Die Daten werden in dem Computer mittels des vordefinierten Dynamic time Warping Algorithmus verarbeitet. Der DTM-Algorithmus wertet die Geste und wählt die zu steuernde Komponente aus. Der Arduino Controller ist sowohl mit dem Computer als auch mit der Komponente verbunden.

4.3 Systementwurf

Die folgende Abbildung zeigt die möglichen Komponenten in der Übersicht.

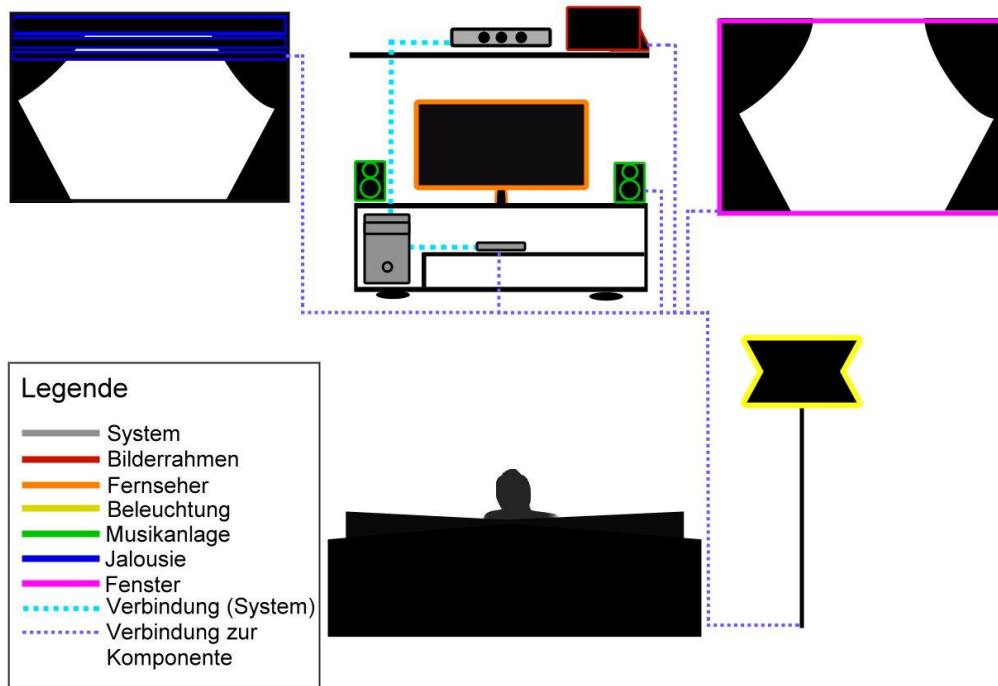


Abbildung 31: Systemübersicht

Das System besteht hardwaretechnisch aus X-Box Kinect, Computer, Arduino Due und den Komponenten

4.4 Software und Sprache

Für die Realisierung wurden verschiedene Programme (Software) und Programmiersprachen verwendet. Mit Software sind integrierte Softwareumgebung (IDE), Bibliotheken (library), Toolkit, Treiber zu verstehen. Die Auswahl der Software und Programmiersprache ist je nach der eingesetzten Hardware mehr oder weniger beschränkt.

4.4.1 WPF Framework

Mit der Einführung von Net-Framework 3.0 wurde Windows Presentation Foundation (WPF) von Microsoft erstmals im Jahr 2006 bekannt. WPF ist eine Library zur Entwicklung von GUI und Multimedia Inhalten. Als Nachfolger von WinForm bietet Windows Presentation Foundation leistungsfähigere und komplexere Benutzeroberflächen zur Entwicklung. Die Grundlage dafür bietet die Extensible Application Markup Language (XAML). Visual Studio ist die geeignete Entwicklungsumgebung von Microsoft. Es stehen dem Entwickler verschiedene

Programmiersprachen zu Verfügung, wie unter anderem C#, C++. Für diesen wissenschaftlichen Beitrag wurde die C# ausgewählt, da die Sprache in der Umgebung am weitesten verbreitet ist.

4.4.2 Processing

Für das Arduino Duo Board bietet sich die Programmiersprache Processing an, die bereits eine integrierte Entwicklungsumgebung hat (siehe Abbildung). Die Sprache ist objektorientiert und die Syntax ist ähnlich wie die Programmiersprache Java. Processing ist Plattform unabhängig in Bezug auf das Betriebssystem und unterstützt verschiedene Micro-Controller. Processing kann für Grafik-, Sound-, Animation-, Typographie, Netzwerkprogrammierung benutzt werden.

4.4.3 Windows Developer Toolkit v1.8

In Windows Developer Toolkit sind verschiedene Source Codes enthalten. Die Source Codes sind programmierte Beispiele, für unterschiedliche Themengebiete. Die Themengebiete umfassen unter anderem Sprachaufzeichnung & Visualisierung, Benutzung von Avatar, Kontrollmechanismen (Controls), Tiefensensor, Facetracking, Skeletal Tracking.

5 Design

In diesem Kapitel wird der Ablauf des Gestenerkennungssystems anhand eines Flussdiagrammes beschrieben. Zudem ist in diesem Kapitel die Erweiterbarkeit des Systems erläutert. Das Design ist für die Realisierung des Gestenerkennungssystems konzipiert.

5.1 Programmablauf

Dieser Abschnitt gibt den theoretischen Ablauf des Programms wieder. Wenn das System gestartet wird, durchläuft das Programm verschiedene Stationen. Die Stationen sind programmiertechnisch die states der „state machine“. Wenn das Programm gestartet wird, gibt es kein Abbruch durch eine Geste. Damit das System beendet wird, muss der entsprechende Task geschlossen werden. Die folgende Abbildung verdeutlicht die einzelnen Stationen.

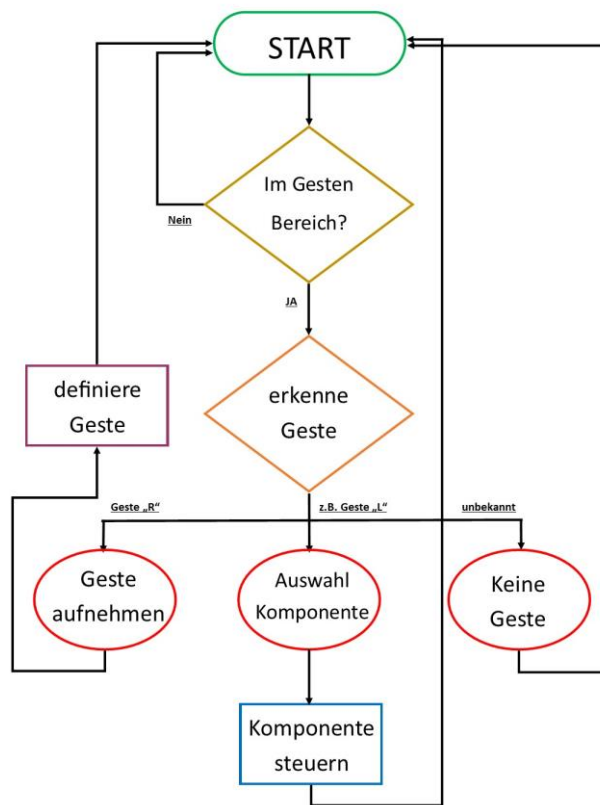


Abbildung 32: Ablauf

Ist das System komplett angeschlossen und das Programm gestartet, befindet sich das Programm zuerst in der Station Start. Befindet sich die Hand anschließend in der richtigen Position, wird die Funktion `erkenne Geste` aufgerufen, die für die Erkennung zuständig ist. Dabei gleicht die Funktion der ausgeführten Geste mit dem Gestenalphabet. Entspricht die ausgeführte Geste keinem Gestenalphabet, so wird das Programm auf Start versetzt. Wird aber die Geste „R“ erkannt, so wird die folgende Geste definiert und einer Komponente zugeordnet. Beim Definieren der Geste ist sicherzustellen, dass keine ähnliche Geste bereits deklariert wurde. Nach der Definition wechselt das Programm auch zur Start Station. Die dritte und letzte Möglichkeit ist, dass eine Geste außer „R“ erkannt wird (Beispielsweise „L“). Zu der erkannten Geste ist immer eine Komponente zugeordnet. Anschließend kann die Komponente reguliert werden. Nach der Regulierung bzw. Steuerung der Komponente wechselt das Programm wieder zu Start.

5.2 Erweiterung des Systems

In dem praktischen Teil der Arbeit wurde anhand der Steuerung vom Licht die Gestenerkennung aufgezeigt. Zudem wurde die Gestenerkennung so entwickelt, dass die Erweiterbarkeit durch weitere Gesten gegeben ist. Die Gesten sind durch die Koordinaten bestimmt, die im System deklariert werden müssen. Theoretisch kann das System durch verschiedene Komponenten erweitert werden. Geräte im Bereich Haushalt, Multimedia, Sicherheit und dergleichen sind einige Beispiele. Entsprechend müssen die Schnittstellen an dem Controller verfügbar gemacht werden und entwickelt werden. Um die Anzahl der physikalisch möglichen Schnittstellen zu erhöhen, besteht die Möglichkeit auf andere Controller zuzugreifen.

6. Realisierung

In diesem Kapitel wird auf die softwaretechnische Entwicklung eingegangen. Der Dynamic-Time-Warping Algorithmus ist in im folgenden Abschnitt erläutert, der für die Erkennung der Gesten zuständig ist. Der folgende Abschnitt gibt einige wichtige Funktionen wieder, die für die Steuerung und Visualisierung der Geste essentiell sind. Alle folgenden Abschnitte geben Einblick in die praktische Umsetzung dieser Arbeit.

6.1 Dynamic-Time-Warping Algorithmus

Der Algorithmus Dynamic-Time-Warping (DTW) wurde ursprünglich für die Spracherkennung entwickelt. Das Verfahren eignet sich, um einzelne Wörter oder Sätze von vorhandenen Sprachaufzeichnungen zu erkennen. Das Verfahren vergleicht die einzelnen Wörter mit den bereits gespeicherten Schablonen (englisch Templates), die meist auf einer Datenbank gesichert sind. Die Sprachsignale werden meist als cepstrale oder spektrale Wertetupel abgesichert. Das Problem ist, dass ein Wort schnell oder langsam ausgesprochen werden kann. Deshalb muss die zeitliche Länge des zu erkennenden Wortes mit der Template abgestimmt werden. Dazu wird das Wort entweder gestaucht oder gedehnt. Dabei ist zu beachten, dass die Konsonanten meist kurzgesprochen werden und die zeitliche Variation oft bei den Vokalen gegeben ist.

Das gleiche Prinzip gilt auch bei der Gestenerkennung. Die Wertetupel sind bei der Gestenerkennung die Koordinaten der Geste. Für die Bewertung der Distanzen werden die Signale gleichzeitig betrachtet (siehe Abbildung). Anschließend berechnet der Algorithmus die Distanz zwischen allen Punkten der beiden Signale. Der Berechnungsaufwand von DTW ist $O(N^2)$ (Salvadore und Chan 2004), wobei N die Anzahl der zu betrachtenden Merkmale entspricht.

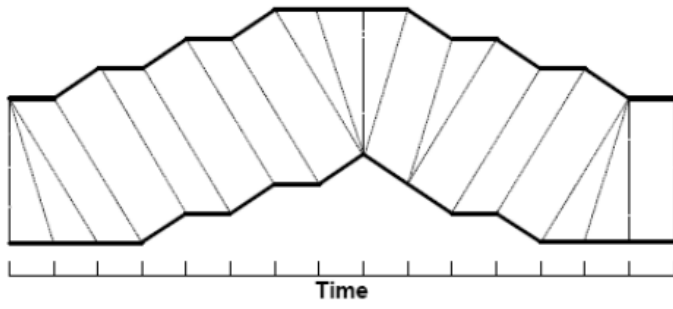


Abbildung 33: Funktionsweise DTW
Quelle: [Fang 2009]

6.2 Kinect Code

Die unten dargestellte Abbildung ist ein Ausschnitt aus der Programmierung. Die Funktion `mySensor_AllFramesReady()` wird erst ausgeführt, wenn der Sensor initialisiert wurde.

```
void mySensor_AllFramesReady(object sender, AllFramesReadyEventArgs e)
{
    ColorImageFrame c = e.OpenColorImageFrame();
    SkeletonFrame s = e.OpenSkeletonFrame();

    if (c == null || s == null) return;

    c.CopyPixelDataTo(myColorArray);
    s.CopySkeletonDataTo(mySkeletonArray);

    //Aufbauen
    BitmapSource bs = BitmapSource.Create(640, 480, 96, 96, PixelFormats.Bgr32, null, myColorArray, 640 * 4);
    DrawingVisual drawingVisual = new DrawingVisual();
    //DrawingContext drawingContext = drawingVisual.RenderOpen();
    drawingContext = drawingVisual.RenderOpen();
    drawingContext.DrawImage(bs, new Rect(0, 0, 640, 480));

    //Rendern
    Pen armPen = new System.Windows.Media.Pen(new SolidColorBrush(System.Windows.Media.Color.FromRgb(255, 0, 0)), 2);
    //Pen legPen = new System.Windows.Media.Pen(new SolidColorBrush(System.Windows.Media.Color.FromRgb(0, 0, 255)), 2);
    Pen spinePen = new System.Windows.Media.Pen(new SolidColorBrush(System.Windows.Media.Color.FromRgb(0, 255, 0)), 2);

    foreach (Skeleton aSkeleton in mySkeletonArray)
    {
        showHandHeadPos(aSkeleton.Joints[JointType.HandRight], aSkeleton.Joints[JointType.Head], drawingContext);
    }

    showDataLabel();
    stateMachine();

    //close
    drawingContext.Close();
    RenderTargetBitmap myTarget = new RenderTargetBitmap(640, 480, 96, 96, PixelFormats.Pbgra32);
    myTarget.Render(drawingVisual);
    imageA.Source = myTarget;

    c.Dispose();
    s.Dispose();
}
```

Abbildung 34: Funktion All Frames Ready

Die unten dargestellte Abbildung, ist ein Ausschnitt aus der Programmierung. Die Funktion `showHandHeadPos()` markiert die Hand und den Kopf des Anwenders.

```
private void showHandHeadPos(Joint handRight, Joint head, DrawingContext aContext)
{
    if (handRight.TrackingState == JointTrackingState.NotTracked || head.TrackingState == JointTrackingState.NotTracked)
    {
        return;
    }
    if (handRight.TrackingState == JointTrackingState.Tracked || head.TrackingState == JointTrackingState.Tracked)
    {
        handRightPos = mySensor.CoordinateMapper.MapSkeletonPointToColorPoint(handRight.Position, ColorImageFormat.RgbResolution640x480Fps30); //HAND
        headPos = mySensor.CoordinateMapper.MapSkeletonPointToColorPoint(head.Position, ColorImageFormat.RgbResolution640x480Fps30); //HEAD
        aContext.DrawEllipse(Brushes.Red, null, new Point(handRightPos.X, handRightPos.Y), 5, 5);
        aContext.DrawEllipse(Brushes.Green, null, new Point(headPos.X, headPos.Y), 3, 3);
    }
} //showHandHeadPos ZU ENDE
```

Abbildung 35: Funktion showHandHeadPos

Die folgende Abbildung zeigt den Aufbau der StateMachine.

```
public enum Status
{
    generateGesture, reconize, setLight, setMusic, showPicture
}

Status currentState = Status.generateGesture;

public void stateMachine()
{
    switch (currentState)
    {
        default:
            case Status.generateGesture:
                generateGestureDefinition();
                currentState = Status.reconize;
                break;

            case Status.reconize:
                recognize();
                showGesturePos(drawingContext);
                break;

            case Status.setLight:
                gestureSetLight();
                break;

            case Status.setMusic:
                gestureSetMusic();
                break;

            case Status.showPicture:
                gestureShowPicture();
                break;
    }
}
```

Abbildung 36: Funktion StateMachine

Die Funktion `waveLengthToRGB` definiert den vertikalen Farbstreifen

```
public int[] waveLengthToRGB(double Wavelength)
{
    double factor;
    double Red, Green, Blue;

    if ((Wavelength >= 380) && (Wavelength < 440))
    {
        Red = -(Wavelength - 440) / (440 - 380);
        Green = 0.0;
        Blue = 1.0;
    }
    else if ((Wavelength >= 440) && (Wavelength < 490))
    {
        Red = 0.0;
        Green = (Wavelength - 440) / (490 - 440);
        Blue = 1.0;
    }
    else if ((Wavelength >= 490) && (Wavelength < 510))
    {
        Red = 0.0;
        Green = 1.0;
        Blue = -(Wavelength - 510) / (510 - 490);
    }
    else if ((Wavelength >= 510) && (Wavelength < 580))
    {
        Red = (Wavelength - 510) / (580 - 510);
        Green = 1.0;
        Blue = 0.0;
    }
    else if ((Wavelength >= 580) && (Wavelength < 645))
    {
        Red = 1.0;
        Green = -(Wavelength - 645) / (645 - 580);
        Blue = 0.0;
    }
    else if ((Wavelength >= 645) && (Wavelength < 781))
    {
        Red = 1.0;
        Green = 0.0;
        Blue = 0.0;
    }
    else
    {
        Red = 0.0;
        Green = 0.0;
        Blue = 0.0;
    }
};

if ((Wavelength >= 380) && (Wavelength < 420))
    factor = 0.3 + 0.7 * (Wavelength - 380) / (420 - 380);

else if ((Wavelength >= 420) && (Wavelength < 701))
    factor = 1.0;

else if ((Wavelength >= 701) && (Wavelength < 781))
    factor = 0.3 + 0.7 * (780 - Wavelength) / (780 - 700);
else
    factor = 0.0;

int[] rgb = new int[3];

rgb[0] = Red == 0.0 ? 0 : (int)Math.Round(IntensityMax * Math.Pow(Red * factor, Gamma));
rgb[1] = Green == 0.0 ? 0 : (int)Math.Round(IntensityMax * Math.Pow(Green * factor, Gamma));
rgb[2] = Blue == 0.0 ? 0 : (int)Math.Round(IntensityMax * Math.Pow(Blue * factor, Gamma));

return rgb;
} //waveLengthToRGB END
```

Abbildung 37: Funktion `waveLengthToRGB`

6.3 Processing und Pololu Lib

Bei der Schnittstellenprogrammierung wurde auf die Pololu Library (Pololu Lib) zugegriffen. Die eingehenden String Werte für Farben werden verarbeitet und an die LED-Beleuchtung weitergeleitet.

```
#include <PololuLedStrip.h>
PololuLedStrip<12> ledStrip;           // Create an ledStrip object and specify the pin it will use.
#define LED_COUNT 60                   // Create a buffer for holding the colors (3 bytes per color).
rgb_color colors[LED_COUNT];

String readString;
String n1, n2, n3, output = "";
int red = 0; int green = 0; int blue = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  while (Serial.available()) {
    delay(3); //delay to allow buffer to fill
    if (Serial.available() > 0) {
      char c = Serial.read(); //gets one byte from serial buffer
      readString += c; //makes the string readString
    }
  }

  if (readString.length() > 0) {
    n1 = readString.substring(0, 3); //get the first three characters
    red = n1.toInt(); //Color red
    n2 = readString.substring(3, 6); //get the second three characters
    green = n2.toInt(); //Color green
    n3 = readString.substring(6, 9); //get the first three characters
    blue = n3.toInt(); //Color blue
    output = "RED: " + (String)red + " " "Green: " + (String)green + " " " + "BLUE: " + (String)blue ;
    Serial.println(output);
    readString = "";
  }

  // Update the colors.
  uint16_t time = millis() >> 2;
  for (uint16_t i = 0; i < LED_COUNT; i++)
  {
    colors[i] = (rgb_color) {
      red, green, blue
    };
  }
  // Write the colors to the LED strip.
  ledStrip.write(colors, LED_COUNT);
  delay(30);
}
```

Abbildung 38: Programmierung der Schnittstelle LED-Strip

6.5 Einblick

In diesem Kapitel sind einige Bilder gemacht wurden, um die praktische Funktionsweise zu zeigen.

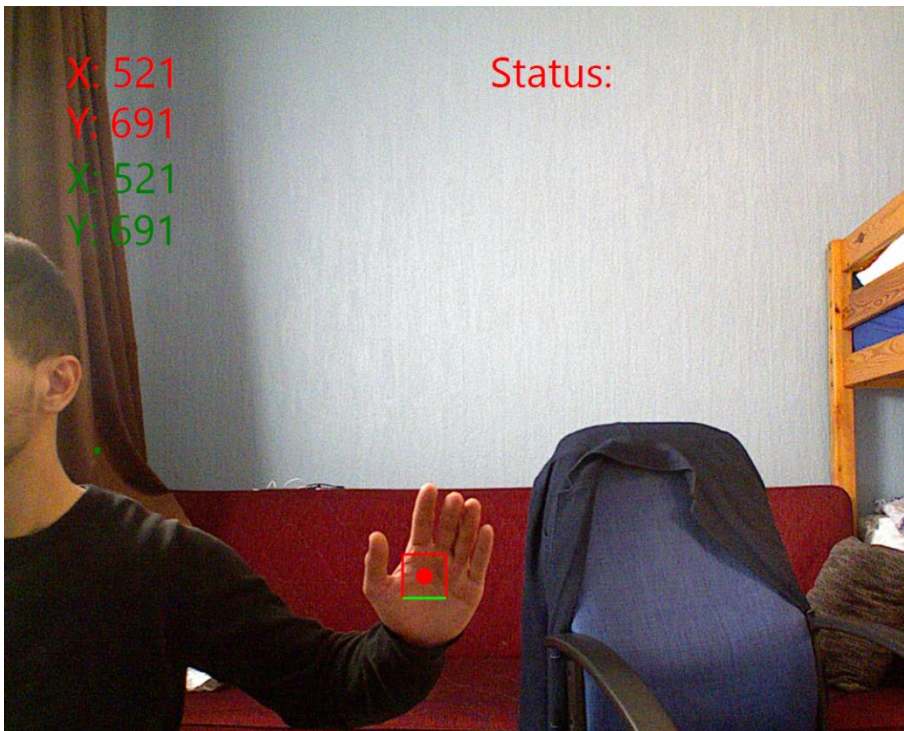


Abbildung 36: Gesture „L“

Die Geste L wird durchgeführt. Der Anwender bewegt die Hand nach unten.

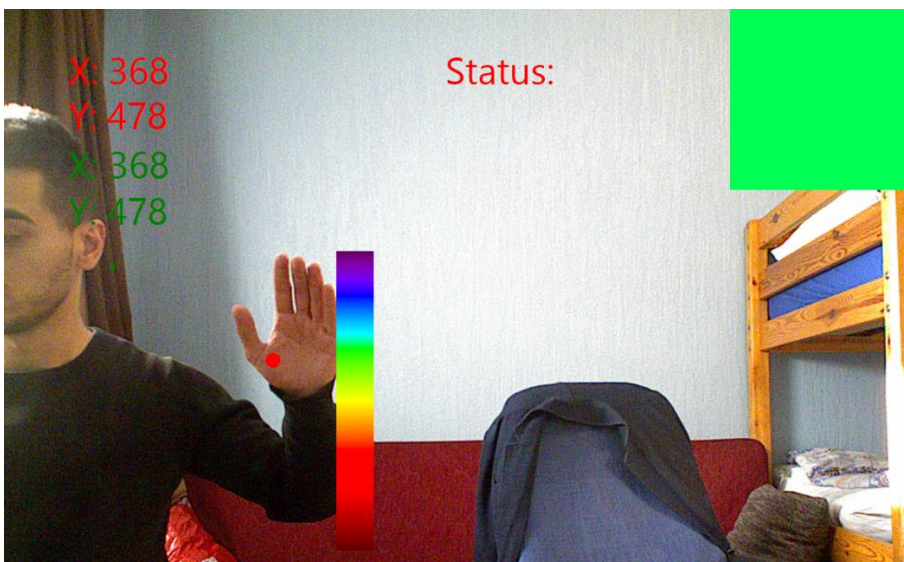


Abbildung 37: chooseColor

Beleuchtung wird reguliert.



Abbildung 38: Farbe auswählen

Nach der Geste „L“ kann die Farbe ausgewählt werden.

7 Schluss

Dieses Kapitel ist der Abschluss der Thesis und beinhaltet die erworbenen Erkenntnisse. Außerdem ist ein Ausblick auf die Entwicklung der Arbeit zu finden.

7.1 Zusammenfassung

Diese Abschlussarbeit stellt einen wissenschaftlichen Beitrag für eine Smart Home Umgebung dar. Der Fokus wurde auf das Konzept der zu steuernden Gesten als auch auf die praktische Entwicklung, um die Mensch Maschine Interaktion (HCI) zu ermöglichen, gelegt.

Zu Beginn der Arbeit wurden grundsätzliche Themengebiete wie die Definition von Geste und Stand der Technik recherchiert. Die Recherche wurde in Kapitel 2 zusammengefasst. Es wurden konkrete Beispiele in den jeweiligen Themengebieten aufgezeigt, um die Grundlagen der Thesis zu verdeutlichen.

Ein Schwerpunkt dieser Arbeit ist die Konzeption der Gestensteuerung. Bei der Erstellung des Konzepts wurden die Ideen des Smart-Home mit berücksichtigt, was im Kapitel 2 erläutert wurde. Die Gesten zur Steuerung wurden in zwei Bereiche unterteilt. Diese sind Gesten zum Auswählen und Gesten zum Manipulieren bzw. zum Einstellen. Zu beiden Bereichen wurden genaue Beispiele aufgezeigt, um das Verständnis des Konzepts aufzuzeigen. Nach der Konzeptionierung wurde intensiv erforscht, welche Hardware und Software eingesetzt werden kann. Die eingesetzte Hardware und Software wurde dargelegt. Anhand eines Flussdiagramms im Kapitel 5 wurde der Ablauf des Systems verbildlicht. Nach der Forschung und der Ausarbeitung des Konzepts wurde mit der Realisierung des Systems begonnen. Bei der Entwicklung wurden die bewährten Technologien ausgewählt. Es wurden auf bereits fertig

gestellte Libraries zugegriffen. Die größten Hürden bei der Realisierung ist die Entwicklung der Gestenerkennung und das Lösen der Start-Stopp-Problematik.

7.2 Ausblick

Diese wissenschaftliche Arbeit ist ein Beitrag zur Entwicklung eines gestenbasierten Steuerungssystems für eine Smart Home Umgebung. In Bezug auf die Steuerung der Computer und Geräte ist ein zu Ende entwickeltes System in der Lage, den Alltag des Menschen zu erleichtern. Dieser Beitrag zeigt das Potential des Systems und die Schwierigkeiten bei der Entwicklung. Die Entwicklung der Erkennung der Gesten ist sehr komplex und aufwendig. Es ist vorteilhaft, die Start-Stopp-Problematik vor der Entwicklung der eigentlichen Gestenerkennung umfassend zu lösen und praktisch umzusetzen, da die Gestenerkennung darauf aufbaut. Wenn die essentielle Entwicklung der Gestenerkennung abgeschlossen ist, sollte das System anschließend mit der Programmierung der Schnittstellen fortgeführt werden, da dann die Grundlage gegeben ist, um das System mit Komponenten zu erweitern.

Im Rahmen der Arbeit wurde versucht, die Schwierigkeiten und die Problematiken, die es zu regeln gilt, zu lösen. Die Probleme sind unterschiedlich wichtig für die Entwicklung des Systems. Die Probleme waren unter anderem das Flackern der LEDs, der Farbunterschied zwischen der angezeigten Farbe am Bildschirm und der tatsächlichen LED-Beleuchtung, das Abstürzen des Programms, die Suchen nach einem angewandten Algorithmus für die Erkennung, das Lösen der Start-Stopp-Problematik und die eigentliche Entwicklung der Gestenerkennung. Viele der Probleme konnte durch Analyse gelöst werden. Eine vollständige Umsetzung der Start-Stopp-Problematik und der Gestenerkennung gilt es noch zu bewerkstelligen.

Das Resultat der praktischen Arbeit ist ein Prototyp, was der weiteren Entwicklung bedarf. Das Konzept wurde ausgearbeitet und Smart Home konform konzipiert. Die Start-Stopp-Problematik und die Gestenerkennung samt DTW Algorithmus müssen vollständig entwickelt werden. Der aktuelle Stand ist, dass per Geste die LED-Beleuchtung gewählt und reguliert werden kann. Analog dazu kann das System mit weiteren Komponenten erweitert werden, wenn der entsprechende Aufwand geleistet wird.

Die Fragestellung, ob die die Gestensteuerung die zukünftige Steuerungsform für die Smart Home Umgebung ist, ist momentan nicht leicht zu beantworten, da die Technologie weiter voranschreiten wird. Wie am Anfang der Arbeit erwähnt, wird in unterschiedlichen Branchen die Gestenerkennung erforscht. Die Realität zeigt, dass neue Steuerungsformen wie per Geste und Sprache immer mehr eingesetzt werden.

Literaturhinweis

[ACM SIGCHI] ACM SIGCHI: Definition and Overview of Human-Computer Interaction.
– URL <http://sigchi.org/cdg/cdg2.html>

[Acm Sigchi Curricula 1992]. Thomas Hewett, Acm Sigchi Curricula for Human Computer Interaction 1992 (schreibweise Quelle)

[Ballmer 2010] BALLMER, Steve: CES 2010: A Transforming Trend – The Natural User Interface.http://www.huffingtonpost.com/steve-ballmer/ces-2010-a-transforming-t_b_416598.html Version: 2010 (Zugriffsdatum 10.10.2015)

[BMW 7er 2015] In allem einen Schritt voraus
<http://www.bmw.de/de/neufahrzeuge/7er/limousine/2015/innovative-funktionen.html>
(Zugriffsdatum 28.11.2015)

[Codeproject Vangos Pterneas 2013]
http://www.codeproject.com/Articles/151605/Kinect-and-WPF-Getting-the-raw-and-depth-image-usi_ (Zugriffsdatum 28.06.2015)

[EECS Unistroke 2014] Electrical Engineering And Computer Science (EECS), Unistroke
www.eecs.yorku.ca/course_archive/2014-15/W/4443/DemoPrograms/demograffiti/Unistroke-Alphabet-Unistrokes.jpg (Zugriffsdatum 14.10.2015)

[Encyclopedia 2003] Encyclopedia (2003). Encyclopedia Americana. Grolier Academic Reference.

[Fang 2009] FANG, Chunsheng: From Dynamic Time Warping (DTW) to Hidden Markov Model (HMM). 03 2009. – University of Cincinnati

[FPV1]http://www.fpv1.de/images/thumbnail/produkte/xlarge/fpv1/elektronik/ws2812B_rgb_led_streifen_weiss_cc3d.jpg (Zugriffsdatum 26.09.2015)

[Goldberg u. Richardson, 1993] Goldberg, D., and Richardson, C., Touch-typing with a stylus. Proc. of INTERCHI93, ACM Press, 1993, S. 80-87

[Inha University 2015] Software Engineering Laboratory, Ubiquitous Computing
– <http://selab.inha.ac.kr/selab2/index.html?p=research/uc> (Zugriffsdatum: 08.07.2015)

[Microsoft Kinect Developer Toolkit 1.8] <https://www.microsoft.com/en-us/download/details.aspx?id=40276>

[Volkswagen 2015] CES (Consumer Electronics Show) 2015 in Las Vegas
http://www.volkswagenag.com/content/vwcorp/info_center/de/themes/2015/01/CES_2015.html (Zugriffsdatum 09.07.2015)

[Pavlovic u. a. 1997] PAVLOVIC, Vladimir I. ; SHARMA, Rajeev ; HUANG, Thomas S.: Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997), S. 677–695

[Pixabay.com 2015] https://pixabay.com/static/uploads/photo/2015/08/01/20/44/volume-870745_960_720.jpg (Zugriffsdatum 09.11.2015)

[PrimeSense Ltd 2010] PRIMESENSE LTD: PrimeSensor Block Diagram.
<http://www.primesense.com/?p=514>. Version: 2010 (Zugriffsdatum 26.09.2015)

[Pololu.com Arduino] Pololu Arduino Controller, Url:
<https://www.pololu.com/product/2193> (Zugriffsdatum 26.09.2015)

[Pololu Lib] Pololu Library Url:<https://github.com/pololu/pololu-led-strip-arduino>

[Salvadore u. Chan 2004] SALVADORE, S. ; CHAN, P.: FastDTW: Toward accurate dynamic timewarping in linear time and space. In: 3rd Workshop on Mining Temporal and Sequential Data 2004

[Strese et al. 2010] Strese, Hartmut ; Seidel, Uwe ; Knappe, Thorsten ; Botthof, Alfons: Smart Home in Deutschland. In: Untersuchung im Rahmen der wissenschaftlichen Begleitung zum Programm Next Generation Media (NGM) des Bundesministeriums für Wirtschaft und Technologie/ Institut für Innovation und Technik (iit) in der VDI/VDE-IT. Institut für Innovation und Technik (2010)

[Teawiking4all 2014] http://www.tweaking4all.com/wp-content/uploads/2014/01/arduino_usb_and_extrenal_power_ws2812.jpg (Zugriffsdatum 28.09.2015)

[Weiser und Brown 1997] Weiser, Mark ; Brown, John S.: Beyond calculation. New York, USA : Copernicus, 1997, Kap. The coming age of calm technolgy, S. 75–85.

[Witura.com Smart Home] http://www.witura.com/images/smart_home_system_1.png
(Zugriffsdatum 11.08.2015)

[Wikimedia.org Apple Newton MessagePad, 2000] Wikimedia
https://upload.wikimedia.org/wikipedia/commons/2/24/Apple_Newton.jpg
(Zugriffsdatum 13.08.2015)

[Wikipedia Wii remote 2014], Wie Remote Url:https://en.wikipedia.org/wiki/Wii_Remote
(Zugriffsdatum 20.11.2015)

[Wobbrock u.a. 2015] University of Washington, Wobbrock 1\$ Recognizer Unistroke
<http://depts.washington.edu/aimgroup/proj/dollar/index.html> (Zugriffzeit 12.11.2015)

[Xbox 360.com] <https://support.xbox.com/de-DE/xbox-360/accessories/kinect-sensor-components>
(Zugriffsdatum 25.09.2015)

[Xbox Kinect 360]
<https://support.xbox.com/de-DE/xbox-360/accessories/kinect-sensor-components>
(Zugriffsdatum 30.10.2015)

[Xbox Kinect 360 Support]
<https://support.xbox.com/de-DE/xbox-360/kinect/kinect-sensor-components>
(Zugriffsdatum 25.09.2015)

[Yuri Arcurs 2009] <https://secure.istockphoto.com/profile/YuriArcurs>
(Zugriffsdatum 07.08.2015)

[Yu Yuan 2008] [Image-based gesture recognition with Support Vector Machines, Yu Yuan
2008

[xSense.com] <https://www.xsens.com/fascination-motion-capture/> (Zugriffsdatum 30.06.2015)

Versicherung über die Selbstständigkeit

Ich versichere, die vorliegende Arbeit selbständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangaben eindeutig kenntlich gemacht.

Ort, Datum

Haci Yigit