

Gestenbasierte Steuerung in Smart Environments

Projekt 2 Ausarbeitung

Sobin Ghose

Wintersemester 2015

Diese Ausarbeitung liefert einen Einblick in die Hauptprojektarbeit in Vorbereitung der Masterarbeit. Dabei wird ein erster Prototyp einer Plattform für HCI Studien zur Gestenerkennung am Beispiel der Steuerung von Haustechnik vorgestellt.

Einführung und Kontext

In der Ausarbeitung des Hauptprojekts sollen die technischen Grundlagen und die Infrastruktur für eine Plattform zur Untersuchung von multimodalen Interfaces zur gestenbasierten Steuerung im Anwendungskontext “Smart Home” konzeptioniert und entwickelt werden. Dabei baut diese Ausarbeitung auf Erkenntnissen der “Anwendungen 1” [6], “Anwendungen 2” [8], “Projekt 1” [9] sowie der “Hauptseminar Ausarbeitung” auf.

1 Motivation und Ziele

Mark Weiser hat durch seine in “The computer for the 21st century” [17] beschriebenen Vision das Bild, wie wir mit Computern umgehen, maßgeblich geprägt. Durch die heutige Rechnerallgegenwärtigkeit (engl. ubiquitous computing) bieten “Natural User Interface” eine intuitive Möglichkeit mit unserer Umgebung zu interagieren. So wird aktuell bereits Sprache zur Steuerung des Autonavigationsgeräts oder des Smartphones genutzt, sowie 2D-Gesten zur Steuerung eines Touchscreens oder 3D-Gesten als Alternative zum Controller bei Spielkonsolen.

Einen weiteren Schritt zur natürlichen Interaktion zwischen Mensch und Maschinen wurde durch Imaginary Interfaces 2010 von P. Baudisch vorgestellt [11]. Diese Konzepte der nahtlosen Interaktion sollen im Smart Home der HAW Hamburg weiter untersucht werden. Dazu wurden bereits erste

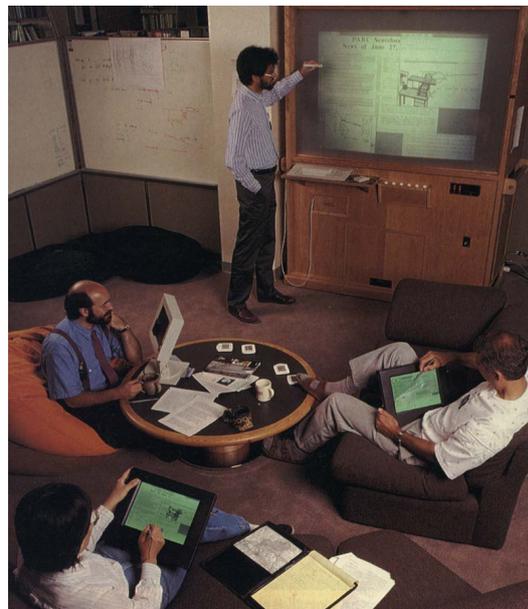


Abbildung 1: Smart Board und Tablets als neue Bedienungskonzepte [17]

Nutzerstudien [3] vor Ort durchgeführt. Um diese Art von HCI Untersuchungen zu erleichtern wurde eine agentenbasierte Plattform konzipiert und entwickelt welche im folgenden vorgestellt wird.

1.1 Vorbereitende Studie

In Zusammenarbeit mit Tobias Eichler wurde ein omnidirektionales Kamera-System zur Erkennung von Skeletten auf Basis von mehreren Kinect 2 Sensoren entwickelt. Dieses System konnte u.a. in einer Kooperation mit der Uni Hamburg verwendet werden. Zusätzlich wurde das System um eine VR Brille erweitert (siehe Abb. 5).



Abbildung 2: Laboraufbau: Der Nutzer trägt eine VR Brille mit einem Notebook im Rucksack. Dabei wird er von vier Kinects v2 getracket [12]

Auf Basis des Motion Tracking Systems und einer weiteren Komponente zur Bodenerkennung konnte eine Schritterkennung durch geführt werden.

Diese Daten, so wie die Ausrichtung und der Beugewinkel des Skeletts wurden als Controller für die Bewegung in der virtuellen Welt genutzt.

Konkret sollte untersucht werden ob eine Beschleunigung in Abhängigkeit des Beugewinkels des Nutzer intuitiv ist und die Immersion erhöht, da man sich beim natürlichen Laufen nach vorne lehnt. Das Paper wurde unter dem Titel "Evaluation of an Omnidirectional Walking-in-Place User Interface with Virtual Locomotion Speed Scaled by Forward Leaning Angle" bei der GIVRAR 2015 Konferenz vorgestellt [12] und hat erste Einblicke in die Problematik der Entwicklung eines Gesten basierten Systems für HCI Untersuchungen geliefert. So konnte durch die Fusion der Skelette eine 360° Erkennung des Nutzers gewährleistet werden, welche nicht mit einem einzelnen optischen Sensors umgesetzt werden kann. Weitere Details zum Tracking System folgen im Kapitel 2.1.2 Fusion Agent oder können dem Paper[12] entnommen werden.



Abbildung 3: Virtuelle Welt: Modell der Hamburg im 9. Jahrhundert [12]

2 Design

Während in der Master Seminar Ausarbeitung [10] bereits intensiv auf die System-Architektur und das Design eingegangen wurden, soll im folgendem auf die konkreten Softwareagenten welche in Abbildung 4 zu sehen sind, sowie auf deren Umsetzung eingegangen werden.

2.1 System-Komponenten

Das Gesamtsystem wurde auf einem frameworkbasierenden System [4] auf Basis von Akka¹ entwickelt. Durch die Struktur des agentenbasierenden Nachrichtenaustausches, wird die Umsetzung der Blackboard Architektur ermöglicht, welche eine hochgradige parallele Bearbeitung verschiedener Aufgaben ermöglicht.

Im folgendem werden die einzelnen Komponenten (siehe Abb. 4) erläutert.

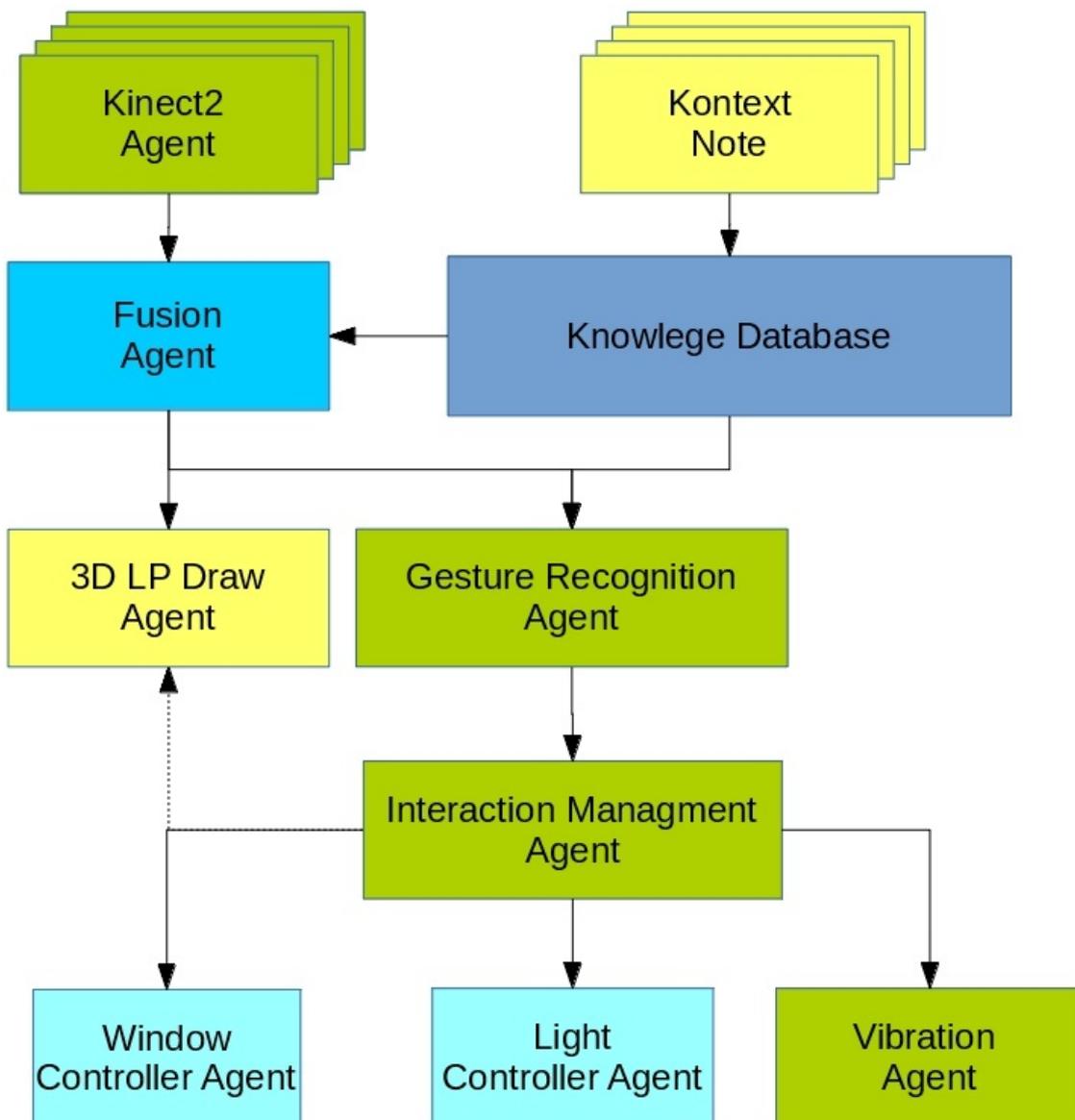


Abbildung 4: Systemkomponenten der multimodalen Plattform

¹Website des Akka-Projekts - <http://akka.io>; letzter Zugriff:04.10.2016

2.1.1 Kinect 2 Agent

Der Kinect 2 Agent übernimmt die Datenabnahme der Kinects und stellt die Skelettinformationen der Plattform zu Verfügung. Der Agent nimmt die Sensordaten einer Kinect an und stellt diese Daten der Plattform über das Blackboard zur Verfügung. Bei der Umsetzung wurde J4KSDK[2] genutzt um durch JNI (Java Native Interface) die Daten aus den C++ Bibliotheken an die JVM(Java Virtual Maschine) weiterzuleiten. Es wird jeweils eine Instanz des Agenten pro Kinect gestartet, daher kann das System auf eine Vielzahl von Kinects skaliert werden. Durch den hohen Datendurchsatz einer Kinect, empfiehlt sich jeweils nur eine einzelne Kinect an einen Computer anzuschließen. Die technischen Rahmenbedingungen des Sensors wurden bereits im Grundprojekt beschrieben[9].

Kontext Note Durch den Umstand das im J4KSDK nicht alle Daten der Kinect 2 mitgeliefert werden musste eine weitere Komponente entwickelt werden, welche dem System Kontextdaten wie die Ebene des Bodens dem System bereitstellt. Diese wird vom Fusion Agent und ggf. vom Gesture Recognition Agent benötigt um den Abstand zum Boden zu messen.

2.1.2 Fusion Agent

Der Fusion Agent soll verschiedene Skelett Daten mit einander Fusionieren (siehe Abb. 5). Dabei setzt sich der Agent aus drei verschiedenen Komponenten zusammen:

1. Zur Kalibrierung der Sensorik und der Umrechnung der verschiedenen 3D-Daten auf ein Weltkoordinatensystem.
2. Zur Berechnung der Transformation der Ebene des Bodens zum Weltkoordinatensystem.
3. Zur Fusion der verschiedenen Skelette durch die in der Kalibrierung erstellten Transformationsmatrizen unter Beachtung verschiedener Randbedingungen zur Sicherstellung eines validen Skeletts.

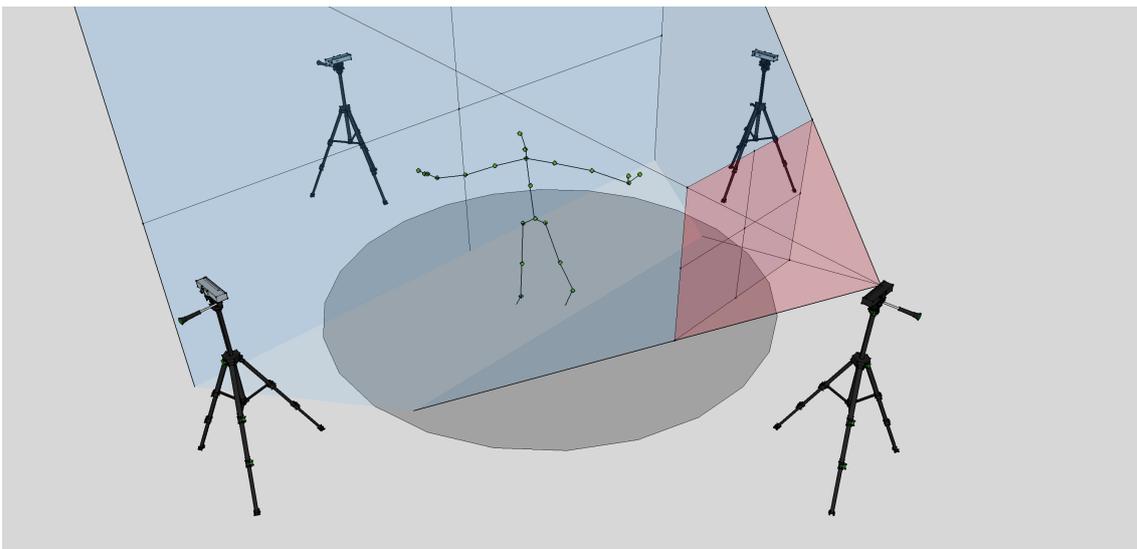


Abbildung 5: Laboraufbau: Während der Nutzer sich bewegt wird dieser von mehreren Kinects erfasst [12]

Vorgehen der Fusion:

1. Die Mittelpunkte der Skelette wird übereinander gelegt.
2. Rotation der Linearen Gleichungssysteme mittels Singulärwertzerlegung(SVD).
3. Sicherstellung verschiedener Randbedingungen welche beispielsweise das Zusammenhängen der Joints garantieren.

Die Skelett Erkennung der Kinect 2 ist lediglich dafür ausgelegt frontal stehende Person zu erkennen. Um so weiter sich eine Person zur Kinect im rechten Winkel stellt, um so schlechter wird die Erkennung. Wenn einer Person mehr als 90° (bzw. weniger als 270°) vor einer Kinect steht, also rückwärtsgewandt, erkennt dies die Kinect nicht und geht von einer frontalen Ausrichtung der Person aus.

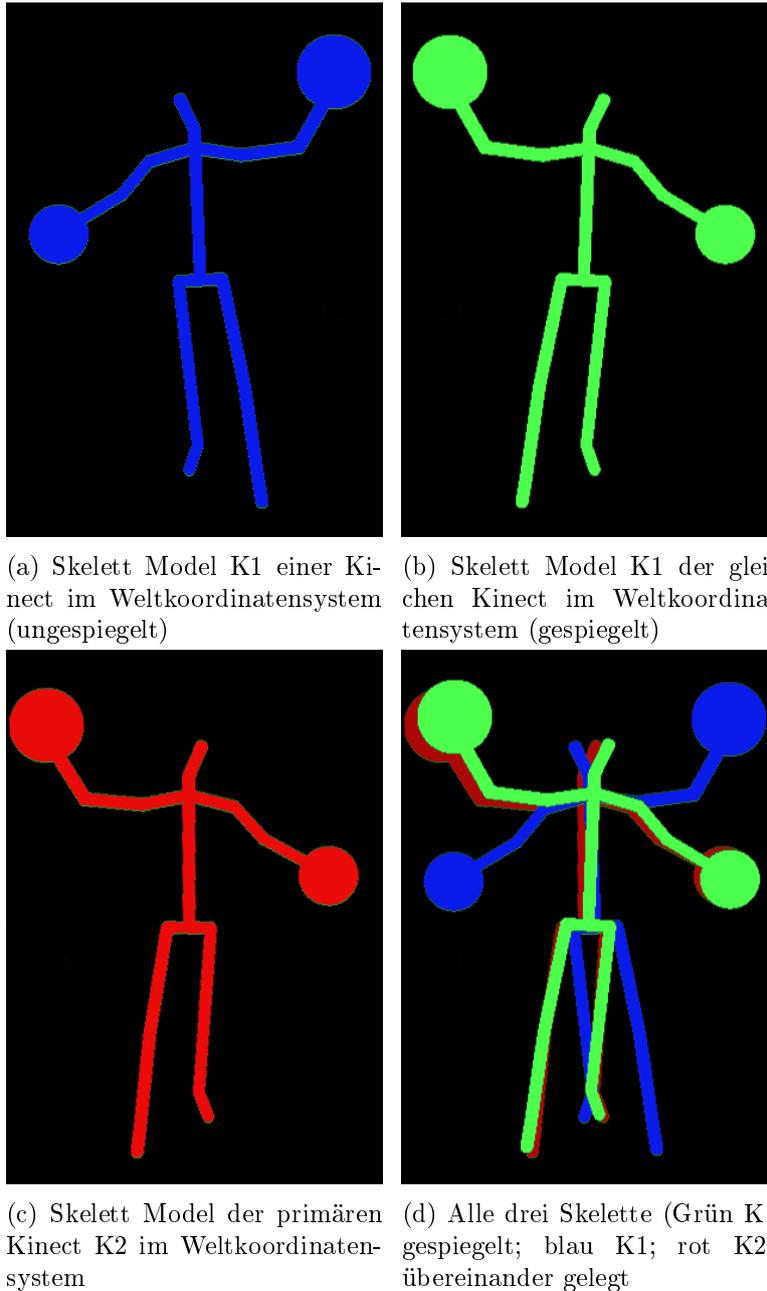


Abbildung 6: Das gespiegelte Skelett (Grün) passt besser zum primären Skelett als das ungespiegelte Skelett. Daher hat die erste Kinect die Person von hinten gesehen, dies aber nicht erkannt.

Um diese Problematiken zu lösen wurden verschiedene Ansätze gewählt. Zum einen wird eine Gewichtung der Skelettdaten anhand des Winkels der Person zur Kinect durchgeführt, dies führt zu weniger Fehlern in der Fusion wenn eine Person im 90° (bzw. 270°) Winkel zu einer oder mehreren Kinects steht. Um die Falscherkennung von rückwärtsgewandten Personen zu verhindern wird für jedes Skelett eine horizontal Spiegelung auf Höhe der Wirbelsäule durchgeführt. Vor der Fusion wird für alle Gelenke der Abstand von den original und den gespiegelten Skeletten zu dem zweiten Skelett berechnet. Dabei wird davon ausgegangen, dass das Skelett bei dem die Summen der Abstände am geringsten ist, richtig gespiegelt ist. Um so unsymmetrischer eine Person steht, um so eindeutiger ist das Ergebnis. Sollte eine Person gerade stehen, ist die Berechnung unzuverlässiger, aber das Ergebnis durch die Symmetrie vernachlässigbar. Dies wird in Abbildung 6 veranschaulicht. Zukünftig soll dieses Verfahren noch durch eine Gesichtserkennung unterstützt werden.

2.1.3 Gesture Recognition Agent

Der Gesture Recognition Agent nimmt das fusionierte Skelette entgegen und erstellt sich pro Skelett einen Hilfsagenten zur Überprüfung auf eine Geste. Dieses gestattet eine hochgradige parallele Berechnung der Gestererkennung so wie eine leichte Verteilung der Rechenlast welche eine gute Skalierung ermöglicht. Weitere Informationen über das verwendete Framework so wie der effizienten Skalierung können aus [4] entnommen werden. Eine Gesten-Bibliothek soll den Entwickler die Implementierung neuer Gesten erleichtern. Dabei besteht eine Geste aktuell aus mehreren Posen, der Entwickler kann neue Posen erstellen und die Anordnung sowie die minimale und maximale Zeit zwischen den Posen bestimmen. In einem ersten Test haben zwei Entwickler auf Basis der Bibliothek bereits mehrere Gesten erfolgreich implementiert.

2.1.4 Interaction Management Agent

Der Interaction Management Agent kennt das anzusteuernde System und enthält die Steuerlogik. Konkret werden aktuell Steuerungsbefehle an Komponenten und System geschickt welche das Licht, die Fenster, die Rollos sowie das Vibrationsarmband verwalten. Damit gibt das Interaction Management den erkannten Gesten erst ihre Bedeutung im Smart Home und ist daher die Schnittstelle zwischen den Sensoren und Aktoren.

Die Steuerbefehle werden entweder durch von anderen Agenten festgelegte APIs definiert, oder als HTTP GET Requests gesendet, welche von anderen in der Wohnung installierten Komponenten interpretiert werden. Aufgrund von fehlenden Dokumentationen mussten diese Requests durch Netzwerkanalysen der Webinterfaces des Pharos Controllers [13] und der Fernbedingung [4] extrahiert werden. Aktuell kann das Interaction Management 80 Trigger per HTTP GET Requests für die Wohnung auslösen und bietet damit umfangreiche Möglichkeiten zur Steuerung des Smart Homes.

2.1.5 Living Place Controller

Für das Living Place existieren verschiedene Agenten und Systeme zur Ansteuerung der Haustechnik wie den Fenstern, den Rollos und Gardinen oder dem Licht (siehe Abb. 7b). Die Ansteuerung dieser Systeme ist bereits in einer erste Implementation des Interaction Management umgesetzt. Da diese Komponenten im Zuge der Überholung des Living Places gegebenenfalls in Kürze ausgetauscht werden, wird an dieser Stelle nicht näher auf die Systeme eingegangen.

Die API definiert lediglich eine Schnittstelle um Vibrations-Muster zu verschicken. Diese bestehen aus zwei Parametern, ein Integer für die Länge eines Vibrationssegments und eine Liste von Booleans die definiert ob im entsprechenden Segment das Armband für die definierte Länge vibriert oder nicht. Der folgende Aufruf würde das Vibrations-Armband beispielsweise zweimal für 120 Millisekunden vibrieren lassen, mit einer Pause von 60 Millisekunden.

```
...vibrate.api.VibratePattern(60, Seq(true,true,false,true,true))
```

Der zweite Prototyp wurde mit dem BLE Nano, einen Bluetooth Low Energy Chip (Nordic nRF51822 SoC [15]) umgesetzt. Dieser ist wesentlich kleiner und der Akku ist mit im Armband verbaut. Die Zuverlässigkeit und die Reichweite der Drahtrosenverbindung konnte durch den Chip Wechsel verbessert werden, zudem ist der neue Chip wesentlich kleiner als ein Arduino LilyPad.



Abbildung 9: kleines BLE Vibrationsarmband

2.2 Design Zusammenfassung

Die Umsetzung des Designs ist größtenteils abgeschlossen, auf noch umzusetzende Agenten und Features wird im Ausblick weiter eingegangen. Erste Tests konnten die Funktionsfähigkeit des Systems auf vertikaler Ebene, also vom Sensor bis zum Aktor zeigen. Durch die Verwendung von BLE statt XBee konnten Antwortzeiten garantiert werden welche die Nutzer in Beta Tests als “unmittelbare Reaktion” wahrnahmen.

Schluss

Im folgenden wird kurz auf die fehlenden Komponenten und weitere mögliche Features eingegangen. Eine detaillierte Evaluation, so wie die Beschreibung der Problemstellungen und Risiken, können der Master Seminar Ausarbeitung [10] entnommen werden.

3 Ausblick

3.1 Face Detection Agent

Um die in Kapitel Fusion Agent 2.1.2 beschriebene Problematik der Frontalerkennung zu lösen kann ein Face Detection Agent implementiert werden um die Ausrichtung des Benutzers festzustellen. In einer ersten Implementierung soll dafür die von Microsoft im Kinect SDK bereitgestellten Funktionen zur Gesichts/Emotionserkennung genutzt werden. Später könnte dieser Agent noch zusätzlich die Gesichtserkennung von OpenCV [16] nutzen, um einen zuverlässigere frontale Erkennung zu gewährleisten.

3.2 Selektion durch Zeige Gestenerkennung

Zudem soll ein Agent implementiert werden, welcher es ermöglicht, Zeige-Gesten wie in Fleeer2012 [5] zu erkennen. Durch diesen sollen dann verschiedene Manipulationen mit dem bereits implementierten Gesture Recognition Agent auf einzelne Fenster oder Lichter umgesetzt werden. Das Vibrationsarmband soll dabei den Nutzer unterstützen die Selektion durchzuführen. Durch das kurze Vibrieren wenn der Nutzer auf ein Objekt zeigt, welches er steuern kann, soll dieser steuerbare Elemente durch das haptische Feedback "erfühlen" können.

3.3 3D Living Place Draw Agent

Die Implementierung der im vorigen Abschnitt erläuterten Erkennung von Zeige-Gesten würde durch die Erstellung eines Draw-Agent in Verbindung mit einem 3D Living Place Modell erleichtert werden. Jedoch muss zuvor die Genauigkeit des vorliegenden Modells so wie der Aufwand für die Kalibrierung ermittelt werden, um eine sinnvolle Aufwand/Nutzen Analyse für ein solches Feature zu erstellen.

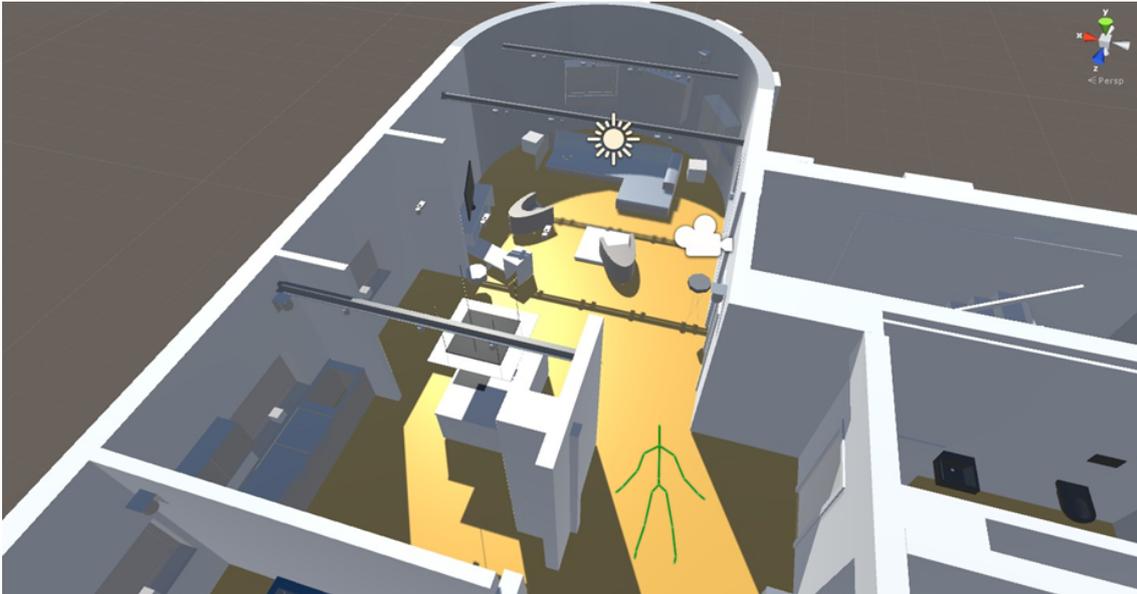


Abbildung 10: Mockup für den Draw Agent

3.4 Selektion und Steuerung durch Spracherkennung

Weiterführend könnte eine Sprachsteuerung über ein Headset, aufbauend auf der in [7] erstellten Grammatik, genutzt werden, um durch Sprachkommandos die Haustechnik anzusteuern. Dies würde auch eine Semantische Fusion von Gestik und Sprache ermöglichen. In einem ersten Schritt würde man durch eine Modalität das Objekt auswählen, und durch die entsprechende andere Modalität das Objekt manipulieren.

4 Fazit

In der Projekt 2 Ausarbeitung konnte die zu Grunde liegende Architektur der Plattform implementiert und getestet werden. Zudem haben bereits erste Entwickler die Plattform genutzt, um eigene Gesten mit den Gesture Recognition Agent zu erkennen. Des Weiteren konnte durch erste Versuche des Vibrations-Armband für ein nutzergerechtes Feedback für Gesten erprobt werden. Das Ziel, ein “Proof of Concept” für eine multimodalen Integrationsplattform für HCI Untersuchungen, konnte durch das Hauptprojekt erfüllt werden.

Literatur

- [1] Dimitra Anastasiou, Cui Jian, and Christoph Stahl. A german-chinese speech-gesture behavioural corpus of device control in a smart home, 2013.
- [2] A. Barmpoutis. Tensor body: Real-time reconstruction of the human body and avatar synthesis from rgb-d. *Cybernetics, IEEE Transactions on*, 43(5):1347–1356, Oct 2013.
- [3] Karolina Bernat. Entwicklung einer gestensteuerung in einer smart-home umgebung. Website, 2015. Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-proj/bernat.pdf>; besucht am 06.09.2015.
- [4] Tobias Eichler. Agentenbasierte middleware zur entwicklerunterstützung in einem smart-home-labor. Website, 2014. Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/eichler.pdf>; besucht am 25.9.2015.

- [5] David Fleer and Christian Leichsenring. Miso: A context-sensitive multimodal interface for smart objects based on hand gestures and finger snaps, 2012.
- [6] Sobin Ghose. Anwendungen 1 Ausarbeitung - Kontextabhaengige Interpretation von 3D-Gesten, 2014.
- [7] Sobin Ghose. Konzeption und evaluation eines interaktiven badezimmerspiegels. Website, 2014. Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/ghose.pdf>; besucht am 06.6.2014.
- [8] Sobin Ghose. Anwendungen 2 Ausarbeitung - Multimodale Haussteuerung, 2015.
- [9] Sobin Ghose. Projekt 1 Ausarbeitung - Multimodale Haussteuerung. Website, 2015. Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-proj/ghose.pdf>; besucht am 13.02.2016.
- [10] Sobin Ghose. Integrationsplattform für HCI Untersuchungen in Smart Environments. Website, 2016. Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-gsem/ghose/bericht.pdf>; besucht am 28.09.2016.
- [11] Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. Imaginary interfaces: Spatial interaction with empty hands and without visual feedback, 2010.
- [12] Eike Langbehn, Tobias Eichler, Sobin Ghose, Gerd Bruder Kai von Luck, and Frank Steinike. Evaluation of an omnidirectional walking-in-place user interface with virtual locomotion speed scaled by forward leaning angle. 2015.
- [13] Pharos Architectural Controls Limited. Lpc. Website, 2016. Erreichbar online unter <https://www.pharoscontrols.com/products/controllers/lpc>; besucht am 06.10.2016.
- [14] Ariza Oscar, Lubos Paul, and Steinicke Frank. Hapring: A wearable haptic device for 3d interaction, 2015.
- [15] Nordic Semiconductor. nRF51822. Website, o.J. Erreichbar online unter <https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF51822>; besucht am 01.11.2016.
- [16] Open Source Computer Vision. Opencv 3.1.0-dev, 2016.
- [17] Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, July 1999.