Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Anwendung 1

Johann Heitsch

Segmentierung von Gesten

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Inhaltsverzeichnis

1 Einführung

Gesten sind neben der Sprache und der Mimik ein wichtiger Bestandteil der menschlichen Kommunikation. Dabei können Gesten die Sprache unterstützen (z.B. die unterstützende Gestik mit Händen im Dialog) oder auch ersetzen, wie es bei der Gebärdensprache der Fall ist. Emotionen, aber auch komplexe Sachverhalte, können über Gesten einfach und intuitiv vermittelt werden. Zudem ist es auch möglich, Befehle und Signale über Gesten zu geben. Ein Beispiel hierfür sind Schiedsrichter im Sport, die über Gesten Entscheidungen bekannt geben.

Es ist zu beobachten, dass Gesten in der Kommunikation eine wichtige Rolle spielen. Ist die Möglichkeit über Gesten zu kommunizieren nicht gegeben, so können schwierige Sachverhalte nur schwer vermittelt werden und es kann schnell zu Missverständnissen kommen.

In der Mensch-Maschine-Kommunikation (HCI[1]) wird seit einiger Zeit der Nutzen von Gesten für die Interaktion erforscht. Ziel ist es, aus der Mensch-Maschine-Interaktion eine Kooperation zu erschaffen. Die Interaktion mit der Maschine soll dabei ähnliche Strukturen annehmen wie die Kommunikation zwischen Menschen.

Die *seamless interaction* ist die Verwirklichung eines mentalen Modells (siehe: Craik (1943)). Die Interaktion findet so statt, wie sich Menschen diese vorstellen. Die Maschinen passen sich den Menschen an und nicht wie bisher die Menschen der Maschine.

Durch die *seamless interaction* (siehe: Ishii u. a. (1994)) verschmelzen die Grenzen zwischen der Informationstechnologie und dem täglichen Leben. Die Forschungen lassen sich somit auch dem Forschungsfeld des *ubiquitous computing*, das von Mark Weiser (Weiser (1991)) geprägt wurde, unterordnen.

1.1 Motivation

Ein Teilgebiet der Mensch-Maschine-Interaktion ist die Interaktion mit großen Displays. Diese Displays weisen Dimensionen auf, welche nicht mehr mit Tastatur und Maus oder Single-/Multitouch Verfahren bedienbar sind. Als Mausersatz kann das Zeigerät gesehen

[1] Human Computer Interaction

werden, welches in Fischer (2008) entwickelt worden ist. Jedoch verändert dieses nicht die Art der Interaktion, sondern nur das Eingabemedium.

Ein neuer Ansatz, welcher auch in dem Film *Minority Report* von Regisseur Steven Spielberg in einer Zukunftsvision dargestellt worden ist, ist der, die Interaktion mittels Gesten zu realisieren. Die Ideen, für die in dem Film gezeigten Interaktionsmodi, stammen vom *Massachusetts Institute of Technology* (MIT (2008)). Viele Entwickler nutzen die Zukunftsvisionen der Filme als Vorlagen, um diese in der Realität zu verwirklichen (vgl. Clarke (2002)).

1.2 Problemstellung

Wie in der Motivation beschrieben, können Gesten die Interaktion zwischen Menschen und Maschinen in speziellen Kontexten erleichtern.

Bei Anwendung der Multitouchtechnologie können 2D Gesten genutzt werden, um Aktionen auszulösen (vgl. Rahimi und Vogt (2008)).

Jedoch ist die Interaktion durch Berührung nicht immer anwendbar. Ein Nachteil ist, dass diese Technologie nicht nutzbar ist, wenn es sich um sehr große Displays handelt. In diesem Fall müsste der Benutzer sehr große Wege zurücklegen, um zu interagieren. Ziel jedoch eine standortunabhängige und berührungslose Interaktion zu ermöglichen.

Es soll daher ein Eingabesystem enwickelt werden, welches unabhängig vom Standort des Benutzers ist und welches Gesten für die Interaktion benutzt.

In der Bachelorarbeit “Ein Framework zur Erkennung von dreidimensionalen Gesten” (Heitsch (2008)) wird beschrieben, wie es möglich ist, beliebige Gesten von Personen zu erkennen. Für die Erkennung des Benutzers wird in der Arbeit ein Motiontracker genutzt.

Wie in der genannten Bachelorarbeit beschrieben wurde, sind für die Erkennung von (isolierten) Gesten definierte Start- und Endpunkte nötig.

Diese Zeitpunkte aus dem Strom von Bewegungsdaten zu extrahieren, ist nicht trivial und das Hauptproblem, mit dem sich diese Ausarbeitung befasst.

Allgemein formuliert ist das Problem somit die Segmentierung von Zeitreihen. Wobei die Zeitreihen die Bewegungsdaten enthalten.

1.3 Ziel dieser Ausarbeitung

Ziel dieser Ausarbeitung ist es, die vorhandenen Methoden zur Segmentierung von Gesten darzustellen und zu vergleichen. Dabei wird das Wissen, welches während der Erstellung der Bachelorarbeit (Heitsch (2008)) gewonnen wurde, genutzt. In dem Kapitel 2.3 wird beschrieben, was nötig ist, um eine System für die Segmentierung zu erstellen.

2 Analyse

Wie in der Einführung beschrieben, müssen die Gesten für die Erkennung segmentiert werden. Bevor jedoch konkrete Überlegungen über die Sementierung getroffen werden können, ist zu evaluieren, wie eine solche Interaktion aussieht und welche Teilbereiche der Interaktion einbezogen werden. Dieses ist in Kapitel 2.1 beschrieben. Eine Auswahl von Algorithmen zur Segmentierung wird im Kapitel 2.2.1 beschrieben und im Kapitel 2.2.2 verglichen.

2.1 Gesten für die Interaktion

Grundsätzlich werden für die Interaktion Gesten benötigt. Dieses sind Bewegungen über die Zeit. Gesten werden von dem Benutzer vollzogen, es ist jedoch nicht festgelegt, mit welchem Körper dieses zu machen ist. So können zum Beispiel Handgesten, wie aber auch alle anderen Formen der Gesten, für die Interaktion genutzt werden.

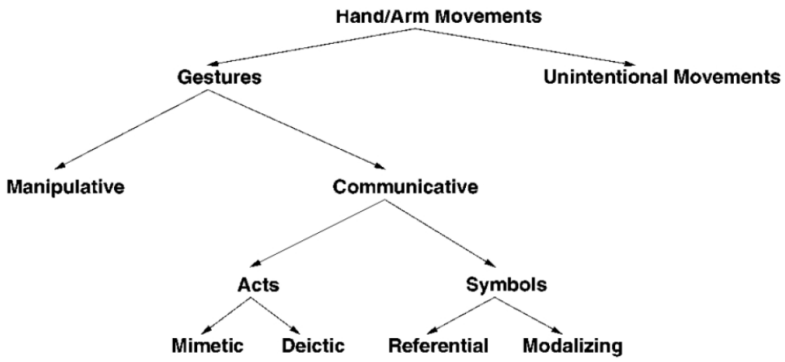

Abbildung 2.1: Taxonomie of Handgestures for HCI (aus Pavlovic u. a. (Jul 1997))

Im allgemeinen können Bewegungen in zwei Klassen eingeteilt werden. Die Taxonomie von Vladimir Pavlovic, Rajeev Sharma und Thomas Huang (Pavlovic u. a. (Jul 1997)), welche in Abbildung 2.1 abgebildet ist, zeigt diese Einteilung. Bei der Segmentierung von Gesten

spielen beide Klassen, sowohl die “Gestures” als auch die “Unintentional Movements”, eine wichtige Rolle.

Die Segmentierung hat die Aufgabe, die Grenzen zwischen diesen beiden Gestenklassen zu finden.

Der manipulative Zweig wird in der Abbildung 2.1 nicht weiter unterteilt, jedoch in der Ausarbeitung über die gestenbasierte Interaktion (J. Boetzer (2008)).

Die für die Interaktion benötigten manipulativen Gesten werden in drei Klassen unterteilt:

Bewegungsverfolgung: Bei dieser Kategorie von Gesten handelt es sich um manipulative Gesten, bei welchen die Bewegung einen direkten Einfluss auf die Position und/oder Orientierung eines Objektes hat. Es können bei der Bewegungsverfolgung nur festgelegte Eigenschaften eines Objektes modifiziert werden. Sie gleichen Gesten aus der Realität.

Kontinuierliche Gesten: Dieses sind Gesten, die in der realen Welt nicht vorhanden sind und vom Benutzer gelernt werden müssen. Es handelt sich bei kontinuierlichen Gesten um eine Mischform aus der Bewegungsverfolgung und den symbolisch-manipulativen Gesten.

Die Objekte werden kontinuierlich modifiziert, jedoch, im Unterschied zu der Bewegungsverfolgung, wird die Art der Modifikaton (die zu modifizierenden Eigenschaften) durch die Bewegung bestimmt.

Symbolisch-manipulative Gesten: Bei diesen Gesten handelt es sich um Gesten, welche vom Benutzer erlernt werden müssen. Es sind Gesten, die Objekte hinsichtlich diverser Eigenschaften verändern können. Für alle Eigenschaften sind eigene Gesten notwendig. Diese Gesten werden benötigt, um Steuerbefehle zu realisieren.

2.2 Segmentierung

Die Segmentierung ist der Prozess der Zerlegung eines Datenstroms in relevante Teilbereiche. In diesem Kontext enthält der Datenstrom die Bewegungsdaten des gestikulierenden Benutzers. Der Datenstrom besteht aus einer Menge von Rohdaten v_i mit ihrer Zeit t_i. Der Vektor beschreibt dabei die Haltung des Benutzers zum Zeitpunkt t_i. Somit lässt sich die Zeitreihe als Skalar $< v_i, t_i >$ schreiben. Zusätzlich gibt es eine konstante Sampelrate k, so dass gilt $t_i + k = t_{i+1}$.

Ein typischer Ablauf eines Segmentierungsprozesses ist in Abbildung 2.2 zu sehen.

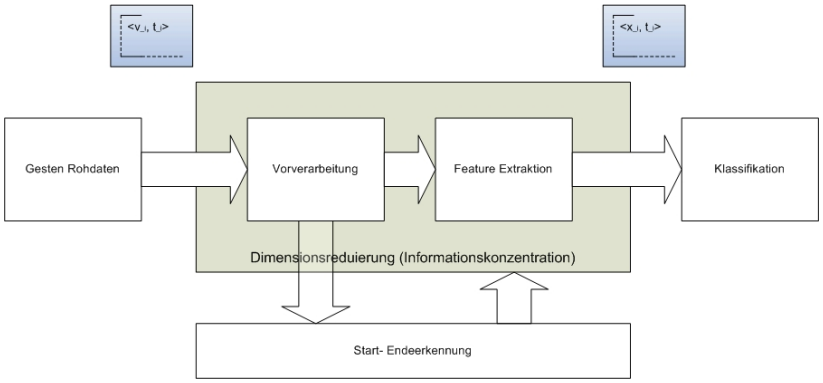

Abbildung 2.2: Ablauf der Gestenerkennung

In der Feature Extraction werden aus den Rohdaten des Datenstoms relevante Features extrahiert. Weitere Informationen hierzu sind Heitsch (2008) zu entnehmen. In der Arbeit werden die Anforderungen genau beschrieben. Die Feature Extraction extrahiert aus den Rohdaten v_i relevante Daten heraus. Diese werden in dem Vektor x_i beschrieben.

Die Segmentierung baut somit auf der Zeitreihe $< x_i, t_i >$ auf. In dieser Zeitreihe sind die Start- und Endpunkte zu identifizieren.

Ein Problem der Segmentierung ist, dass für die Lösung des Segmentierungsproblems bereits die Lösung des Gesamtproblems benötigt wird. Auf diese Problematik wird in 2.3 eingegangen.

2.2.1 Algorithmen

Im Folgenden werden einige Algorithmen beschrieben, mit denen es möglich ist, eine Segmentierung durchzuführen. Der Fokus der Beschreibungen liegt dabei auf den groben Konzepten und nicht auf den Details.

Es wird zudem darauf eingegangen, für welche Art von Gesten aus Kapitel 2.1 sich die Algorithmen anwenden lassen, wobei die Gesten der *Bewegungsverfolgung* nicht betrachtet werden. Die Gesten müssen nicht erkannt werden, sondern die Bewegungen werden direkt auf Objekte angewendet.

Unterteilt werden die Algorithmen in ein- und zweistufige Verfahren. Bei den einstufigen Verfahren wird die Segmentierung und die Klassifizierung in einem Schritt durchgeführt, wobei im zweistufigen Verfahren die Segmentierung von der Klassifizierung getrennt ist. Die Klassifizierung für das zweistufige Verfahren ist bereits in Heitsch (2008) beschrieben und gelöst worden.

Zweistufige Verfahren

Sliding Window Bei der Segmentierung mittels des *Sliding Window*- Verfahren wird jede mögliche Segmentierung betrachtet. Es wird dabei das Sliding Window w definiert, welches die Breite b besitzt. Dieses Fenster wird über den Datenstrom gelegt und kontinuierlich um s Schritte verschoben. Hat das Fenster eine neue Position erreicht, so werden die Daten in dem Fenster der Klassifizierung zugeführt und ermittelt, ob es sich um eine Geste handelt.

Vorraussetzung für diese Verfahren ist, dass der Klassifizierer die Güte der Erkennung liefern kann. Anhand dieser kann bestimmt werden, ob es sich tatsächlich um eine Geste gehandel. Der Vorteil dieser Methode ist, dass auf jeden Fall jede Geste erkannt wird.

Der Nachteil hingegen ist, dass dieses Verfahren zu langsam für Echtzeitanwendungen ist.

Mit dem *Sliding Window*- Verfahren können *Symbolisch-manipulative Gesten* erkannt werden. *Kontinuierliche Gesten* können auch erkannt werden, wobei sich hier die Erkennung mehr auf das Erkennen von Posen, welche die Art eine Modifikation beschreibt, fokussiert.

Regelbasiert Ein anderer Ansatz, ist es die Segmentierung statisch und nicht stochastisch durchzuführen. Es sind Regeln nötig, welche fest definiert werden müssen. Hierzu muss ein Merkmal bestimmt werden, welches essentiell für eine Geste ist. Als Beispiel könnte die Bewegung der Hand genommen werden oder eine festgelegte Pose der Hand. Es werden mindestens zwei Kriterien benötigt, eine für den Startpunkt der Geste und eine für den Endpunkt einer Geste. Die Kriterien beschreiben hierbei, wie sich das Merkmal verhalten muss, um das Kriterium zu erfüllen.

Der Vorteil dieser Methode ist, dass die Implementierung sehr einfach ist und der Rechenaufwand gering ist. Jedoch ist es nicht einfach geeignete Kriterien zu definieren.

Mit dem *Regelbasierten*- Verfahren können *Symbolisch-manipulative Gesten* erkannt werden. *Kontinuierliche Gesten* können auch erkannt werden, wobei mittels Kriterien beschrieben werden muss, um was für eine Art von Modifikation es sich handelt.

Vorgabe Ein auf dem vorgabebasierten Ansatz aufbauender Ansatz ist der, der auch in der Bachelorarbeit Heitsch (2008) beschrieben wurde. Anstatt das System die Segmentierung vornehmen zu lassen, werden die Start- und Endpunkte einer Geste vom Benutzer selbst vorgegeben.

Der Vorteil dieser Methode ist, dass keine wirkliche Segmentierung benötigt wird und somit der Rechenaufwand vernachlässigbar gering ist. Jedoch muss der Benutzer ein weiteres Eingabegerät nutzen, um die Segmentierung zu steuern.

Mit dem *Vorgabebasierten-* Verfahren können *Symbolisch-manipulative Gesten* erkannt werden. *Kontinuierliche Gesten* können auch erkannt werden, wobei über die Art der Vorgabe (je eine Taste pro Modifikation) bestimmt wird, um was für eine Art von Modifikation es sich handelt.

Dynamische Programmierung Bei der Segmentierung mittels *Dynamischer Programmierung* wird eine sehr kritische Segmentierung vorgenommen. Der Bewegungsablauf wird somit in sehr viele kleine Abschnitte zerteilt. Nach der Zerteilung wird für jeden Abschnitt bestimmt, wie groß die Wahrscheinlichkeit ist, dass es sich dabei um eine komplette Geste handelt. Diese Wahrscheinlichkeit wird auch für benachbarte Abschnitte bzw. für Gruppen von benachbarten Abschniitten berechnet. So ergibt sich ein Graph, in dem mittels dem Viterbi- Algorithmus der Weg mit der maximalen Wahrscheinlichkeit berechnet werden kann. Die Abschnitte, die auf diesem Weg liegen, sind die gesuchte Geste.

Wie auch bei dem *Sliding Window-* Verfahren ist ein Klassifizierer nötig der die Güte der Erkennung und somit die Wahrscheinlichkeit berechnen kann, dass es sich bei dem Abschnitt um eine Geste handelt.

Das Verfahren ist jedoch sehr langsam und rechenaufwendig und ist für Datenströme eher ungeeignet.

Mit der *Dynamische Programmierung* können *Symbolisch-manipulative Gesten* erkannt werden. *Kontinuierliche Gesten*, wie in Heitsch (2008) beschrieben, können nicht erkannt werden, da die Art der durchzuführenden Modifikation vor dem Ende der Geste bekannt sein muss.

Einstufige Verfahren

Hidden Markov Model Bei einem *Hidden Markov Model* handelt es sich um ein stochastisches Modell, welches zwei Zufallsprozesse beinhaltet und schon seit langem in der Spracherkennung erfolgreich eingesetzt wird.

Der erste Zufallsprozess entspricht dabei einer *Markov- Kette*, welche Zustände und Übergangswahrscheinlichkeiten enthält. Ein HMM ist eine Zustandsmaschine, bei der die Zustände nicht direkt sichtbar (*hidden*) sind. Der Zustand des HMM ist, über die Ausgangssymbole, die in jeden Zustand ausgegeben werden, beobachtbar. Ein Zustand besitzt dabei nicht ein stisches Ausgangssymbol, sondern für jedes Ausgangssymbol eine Wahrscheinlichkeit, dass dieses ausgegeben wird.

Ein HMM ist formal ein Quintupel $\lambda = (S, A, B, \pi, V)$ mit

- $S = s_1, \ldots, s_n$ - Menge aller (versteckten) Zustände
- $A = a_{ij}$ - Zustandsmatrix, wobei a_{ij} die Übergangswahrscheinlichkeit von Zustand a_i zum Zustand a_j angibt
- $B = b_i, \ldots, b_n$ - Menge der Ausgabewahrscheinlichkeiten
- $b_i(x)$ - Wahrscheinlichkeit im Zustand s_i die Beobachtung $x \in V$ zu machen
- π - Anfangswahrscheinlichkeitsverteilung mit $\pi(i)$ Wahrscheinlichkeit, dass s_i der Startzustand ist
- V - Ausgabealphabet

Es handelt sich bei diesem Verfahren um ein lernendes Verfahren. Die Übergangs- und Ausgabewahrscheinlichkeiten müssen nicht vorgegeben werden, sondern werden mittels Trainingsdaten dem System mit dem *Baum- Welch- Algorithmus* antrainiert. Um festzustellen, ob es sich bei der Eingabesequenz um eine angelernte Geste handelt, ist folgendes Problem zu lösen: Für ein gegebenen HMM λ ist die Wahrscheinlichkeit zu berechnet, dass die Beobachtung O eine Ausgabefolge des HMM ist. Formal bedeutet dieses: $P(O|\lambda)$. Für die Berechnung wird der *Viterbi- Algorithmus* genutzt.

Die Berechnung der Wahrscheinlichkeit $P(O|\lambda)$ wird ähnlich, wie im *Sliding Window*-Verfahren, mit jedem neuen Datensatz weitergeführt und gestartet. Mit jedem neuen Datensatz werden alle HMM mit dem *Viterbi- Algorithmus* geprüft. Am Ausgang des HMM ist jeweils die Wahrscheinlichkeit zu entnehmen. Überschreitet dieser einen Grenzwert, so ist eine Geste erkannt. Veringert sich dieser, so kann die Instanz des Viterbi- Algorithmus gestopt werden.

Der Vorteil von HMM ist, dass die Gesten über ein Training leicht erlernt werden können. Zu jeden Zeitpunkt ist bekannt, welche Geste zur Zeit am Wahrscheinlichsten durchgeführt wird. Somit kann eine Geste schon erkannt werden, bevor diese abgeschlossen ist. Zudem ist der Implementationsaufwand an ein Gestenerkennungssystem und die Fehleranfälligkeit geringer, da insgesamt weniger Algorithmen benötigt werden.

Der Nachteil ist jedoch, dass viel und gutes Trainingsmaterial benötigt wird. Der *Viterbi-Algorithmus* muss zudem an die kontinuierliche Erkennung angepasst werden.

Mit den HMMs lassen sich auch die *Symbolisch-manipulative Gesten* erkennen. Dadurch, dass sehr schnell ein Ergebnis über die Ausführung einer Geste vorhanden ist, kann es möglich sein, mit den HMM auch *Kontinuierliche Gesten* zu erkennen.

2.2.2 Vergleich

Im vorherigen Kaptitel wurden einige Algorithmen zur Segmentierung von Gesten vorgestellt. Um einen Lösungsansatz (Kapitel 2.3)zu beschreiben ist es jedoch nötig, die Konzepte zu vergleichen. Im Folgenden werden zunächst die Kriterien für einen Vergleich bestimmt und danach die Algorithmen verglichen.

Kriterien

Die Algorithmen sollen in folgenden Punkten verglichen werden:

- Rechenaufwand - Wieviel Rechenaufwand ist nötig, um den Datenstom zu segmentieren?
- Ergebniszeitpunkt - Wann ist das Ergebnis der Gestenerkennung für Symbolisch-manipulative Gesten vorhanden?
- Erkennung der Gestenklassen - Welche Klassen von Gesten können segmentiert werden (S: Symbolisch-manipulative Gesten, K: Kontinuierliche Gesten)?

Vergleich

Algorithmus	Rechenaufwand	Ergebniszeitpunkt	Gestenklassen
Sliding Window	hoch	spät	S + K
Regelbasiert	gering	spät	S + K
Vorgabe	0	spät	S + K
Dynamische Programmierung	hoch	spät	S
Hidden Markov Model	gering	früh	S + K

In der Tabelle werden die Algorithmen nach den oben genannten Kriterien verglichen.

Auffällig ist, dass die *Dynamische Programmierung* sich schlecht für die *Kontinuierliche Gesten* einsetzen lässt, zudem eine hohe Laufzeit hat und das Ergebnis erst spät, also nach Beendigung der Geste, vorhanden ist.

Die zweistufigen Verfahren wie das *Sliding Window*-, und *Regelbasierte*- Verfahren können zwar Gesten aller Klassen erkennen, jedoch ist das Ergebnis jeweils auch erst nach der vollzogenden Geste bekannt.

Am besten eignen sich das *HMM*. Die Ergebnisse sind früh vorhanden, der Rechenaufwand ist gering und beide Gestenklassen lassen sich erkennen.

Gutes Trainingsmaterial ist für alle Algorithmen nötig. Bei den *HHM* wird es sowohl für die Segmentierung, als auch für die Erkennung benötigt. Die zweistufigen Verfahren benötigen Trainingsmaterial für die Erkennung von den isolierten Gesten.

2.3 Lösungsansatz

Der Vergeich der Algorithmen alleine reicht nicht aus, um eine Aussage darüber zu treffen, welcher Algorithmus am besten geeignet ist, die Gestensegmentierung zu übernehmen. Abhängig ist dieses sehr stark von dem Szenario und der Anwendung, welche bedient werden soll.

Abhängig vom Szenario sind *Symbolisch-manipulative Gesten* oder *Kontinuierliche Gesten* nötig. Für das Verschieben, Rotieren, Skalieren und ähnliche Manipulationen an Objekten, auf z.B. Powerwalls, sind die *Kontinuierlichen Gesten* notwendig. Anhand festgelegter Merkmale wird bestimmt, um was für eine Manipulation handelt. Diese wird dann durch die Bewegung vollzogen.

In einem Kontext, wie z.B. dem *Ambient Living*, bei dem weniger Objekte manipuliert werden, sondern Aktionen, wie z.B. das Licht einschalten, per Gesten gesteuert werden sollen, sind *Symbolisch-manipulative Gesten* nötig. Im Kontext des *CSCW*s sind jedoch *Kontinuierliche Gesten* von größerer Bedeutung.

Jedoch sind dieses eher grobe Annahmen. Um herauszufinden, was für Gesten in welchem Kontext von Benutzern gefordert werden, ist es nötig, ähnlich wie in Morguet (2000) beschrieben, eine Usabilitystudie durchzuführen. Dazu ist es notwenfig eine Testumgebung zu erschaffen, in der Testbenutzer vor Aufgaben gestellt werden, die sie mit Hilfe von Gesten zu erledigen haben. Im ersten Teil der Studie geht es darum, Informationen zu sammeln, wie Benutzer mittels Gesten interagieren. Das Vorgehen kann sehr einfach wie folgt beschrieben werden: Einem Benutzer wird auf der Powerwall ein einfaches Programm, zum Beispiel ein Bilderpuzzle gezeigt. Anschließend wird der Benutzer aufgefordert, eine Modifikation, wie das Verschieben oder Drehen von Puzzleteilen, durrchzuführen. Hierbei soll der Benutzer jede Aktion, die er macht, mit Worten begleiten. Eine Person an dem Steuerungs PC gibt die Eingaben in das System ein und modifiziert damit die aktuelle Szene. Es ist noch kein System zur Gestensegmentierung oder Erkennung nötig.

Dem Testbenutzer wird nicht vorgeschrieben, auf welche Art und Weise er mit der Anwendung zu interagieren hat. Er kann jede, die ihm zur Verfügung stehende, Technik nutzen.

Während der Benutzer mit dem System interagiert, muss dieser mit Kameras gefilmt werden. Es ist notwenig einen Fragenkatalog zu erarbeiten, welcher von der Testperson nach dem Versuch auszufüllen ist.

Die Fragen in dem Fragenkatalog müssen darauf abgestimmt sein, herauszufinden, ob die Art der Interaktion dem Benutzer angenehm ist, oder ob die Gesten auf Dauer zu anstrengend sind und von der eigentlichen Arbeit ablenken.

Anhand der gesammelten Daten kann nun entschieden werden, was für ein System sich am besten für die Interaktion mit der Anwendung eignet.

3 Auswertung

3.1 Ausblick

Um ein Eingabesystem zu realisieren, welches die Segmentierung und Erkennung automatisch vornimmt, ist es nötig, die Usabilitystudie, die in dem Lösungsansatz beschrieben ist, auszuarbeiten und durchzuführen. Anhand der Ergebnisse ist ein System zu implementieren, welches den Anforderungen aus der Usabilitystudie gerecht wird. Anschließend ist eine Evaluation des Systems nötig.

3.2 Risiken

Die größten Risiken bestehen in der Usabilitystudie. Diese muss Ergebnisse liefern, anhand denen die Auswahl eines Algorithmusses möglich ist. Zudem müssen nach der Studie ausreichend viele Trainingsdatensätze vorhanden sein, mit denen ein System trainiert werden kann.

Ein weiteres Risiko stellt der Benutzer an sich da. Wenn der Benutzer nicht mit dem System zurecht kommt oder dieses nicht akzeptiert, so hat das System keinen Nutzen.

Literaturverzeichnis

Clarke 2002 CLARKE, Darren J.: *MIT grad directs Spielberg in the science of moviemaking*. Webseite. July 2002. – URL http://web.mit.edu/newsoffice/2002/underkoffler-0717.html. – Zugriffsdatum: 12.08.2008

Craik 1943 CRAIK, K.: *The Nature of Explanation*. Cambridge University Press, 1943

Fischer 2008 FISCHER, Christian: *Entwicklung eines multimodalen Interaktionssystems für computergestützte Umgebungen*, HAW Hamburg, Masterarbeit, 2008. – URL http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/fischer.pdf. – Zugriffsdatum: 12.08.2008

Heitsch 2008 HEITSCH, Johann: *Ein Framework zur Erkennung von dreidimensionalen Gesten*, HAW Hamburg, Bachelorarbeit, 2008. – URL http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/heitsch.pdf. – Zugriffsdatum: 12.08.2008

Ishii u. a. 1994 ISHII, Hiroshi ; KOBAYASHI, Minoru ; ARITA, Kazuho: Iterative design of seamless collaboration media. In: *Commun. ACM* 37 (1994), Nr. 8, S. 83–97. – ISSN 0001-0782

J. Boetzer 2008 J. BOETZER, M. Vogt P. Wendt K.: Gestenbasierte Interaktion mit Hilfe von Multitouch und Motiontracking. In: CLEVE, Jürgen (Hrsg.): *WIWITA 2008* Hochschule Wismar (Veranst.), Hochschule Wismar, 2008, S. 38–49

MIT 2008 MIT: *Massachusetts Institute of Technology*. Webseite. 2008. – URL http://web.mit.edu/. – Zugriffsdatum: 12.08.2008

Morguet 2000 MORGUET, Peter: *Stochastische Modellierung von Bildsequenzen zur Segmentierung und Erkennung dynamischer Gesten*. München, Technische Universität München, Dissertation, 2000

Pavlovic u. a. Jul 1997 PAVLOVIC, V.I. ; SHARMA, R. ; HUANG, T.S.: Visual interpretation of hand gestures for human-computer interaction: a review. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19 (Jul 1997), Nr. 7, S. 677–695. – ISSN 0162-8828

Rahimi und Vogt 2008 RAHIMI, Mohammadali ; VOGT, Matthias: *Gestenbasierte Computerinteraktion auf Basis von Multitouch-Technologie*, HAW Hamburg, Bachelorarbeit, 2008

Weiser 1991 WEISER, Mark: The Computer for the 21st Century. In: *Scientific American* (1991), Februar. – URL http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html. – Zugriffsdatum: 12.08.2008