

# Blickgesten als Fernbedienung

Heiko Drewes<sup>1</sup>, Heinrich Hußmann<sup>1</sup>, Albrecht Schmidt<sup>2</sup>

<sup>1</sup>Institut für Informatik, LMU München,

<sup>2</sup>Fraunhofer IAIS und B-IT Universität Bonn

## Zusammenfassung

Dieser Beitrag stellt Blickgesten als ein neues Konzept für die Bedienung von Computersystemen mit den Augen vor. Blickgesten sind immun gegen Kalibrierungsdrift und unempfindlich für Genauigkeitsfehler. Sie eignen sich besonders für Augensteuerung aus der Entfernung. Es konnte gezeigt werden, dass Menschen in der Lage sind, willentlich kontrollierte Gesten mit dem Blick auszuführen. Des Weiteren wurde ermittelt, welchen Grad an Komplexität die Gesten besitzen müssen, um von natürlichen Augenbewegungen unterschieden werden zu können. Die vorliegende Arbeit erläutert den Gesten-Algorithmus und dessen Implementierung, präsentiert die Ergebnisse und Analysen zweier Benutzerstudien und diskutiert mögliche Anwendungen der Blickgesten.

## 1 Einleitung

Praktikable Blickverfolgersysteme, auch Eye-Tracker genannt, als Beobachtungswerkzeuge für psychologische Studien gibt es seit den 70er Jahren. Die ersten Untersuchungen von Blickverfolgern zur Steuerung von Computern wurden Anfang der 80er bis Anfang der 90er durchgeführt (Bolt 1981; Ware & Mikaelian 1987; Jacob 1990). Seit dieser Zeit gibt es Blickverfolgersysteme als Eingabegeräte, die es behinderten Menschen, die nichts außer ihren Augen bewegen können, ermöglicht, Texte am Computer einzugeben (Majaranta & Rähä 2002). Diese Systeme basieren auf der Erkennung der Pupille in einem Kamerabild. Blickverfolger, die die freie Bewegung vor dem Bildschirm erlauben, verfolgen zusätzlich die Position und Orientierung des Kopfs, was ebenfalls über die Auswertung eines Kamerabilds erreicht wird. Diese Systeme sind noch teuer, aber als kommerzielle Produkte erhältlich, z.B. der EyeGaze EyeFollower<sup>1</sup> oder Tobii 1750 Eye Tracker<sup>2</sup>. In den letzten Jahren sind die Preise für digitale Kameras drastisch gesunken und eine günstige Produktion von

---

<sup>1</sup> <http://www.eyegaze.com/>

<sup>2</sup> <http://www.tobii.com/>

Blickverfolgern für den Massenmarkt wäre möglich. Die Frage ist jedoch, welche Einsatzbereiche in der Bedienung von Computern für Blickverfolgersinnvoll sein können, insbesondere für Menschen die alternative Modalitäten verwenden können. Zum Beispiel ist der Einsatz von Blickverfolgern zur Eingabe von Text wenig effektiv im Vergleich zur Tastatur.

Die Verwendung des Blicks als Eingabemodalität bringt spezifische Probleme mit sich, die nicht alleine in der Technik liegen, sondern in den Augenbewegungen selbst. Das Auge bewegt sich in abrupten Sprüngen (Sakkaden) und verharrt einige Zeit an einer Position (Fixationen). Während einer Fixation führt das Auge kleine Zitterbewegungen (Nystagmus) und Mikrosakkaden aus. Diese Unruhe des Auges begrenzt die erreichbare Genauigkeit unabhängig von Messgenauigkeiten. Kommerziell erhältliche Systeme geben ihre Genauigkeit mit  $\pm 0,5^\circ$  an. Mit heutigen Bildschirmauflösungen führt dies zu Blick-Interaktionsobjekten mit Abmessungen von mindestens 40 Pixeln. Ein weiteres Problem besteht in der Tatsache, dass der Blick nicht wie ein Finger auf- und abgesetzt werden, und somit keine Knöpfe drücken kann. Um eine Aktion mit dem Blick auszulösen muss das Bedienelement eine bestimmte Zeit lang angestarrt werden (Dwell-Time-Methode). Der Zeitvorteil, der durch die sprichwörtlich schnellen Augenbewegungen erreicht werden könnte, wird so wieder zunichte gemacht. Des Weiteren sind die Augen in erster Linie zum Sehen und nicht für das Bedienen von Computern gemacht. Eine ausgiebige Verwendung der Augen zum Steuern könnte mit der primären Funktion des Sehens in Konflikt geraten.

Zur Lösung des Genauigkeitsproblems wurden verschiedene Vorschläge gemacht. Ein Ansatz besteht darin, dass sich die Bedienelemente beim Ansehen vergrößern (Špakov & Miniotas 2005). Ein anderer Ansatz ist die Auswahl des richtigen Bedienelements in der Nähe der Blickposition durch intelligente Algorithmen (Salvucci & Anderson 2000). Zu den möglichen Konflikten bei der Verwendung des Blicks als Ein- und Ausgabekanal gibt es verschiedene Ansichten. Manche Wissenschaftler sehen die Verwendung des Blicks zur expliziten Steuerung eher skeptisch (Zhai et al. 1999) und schlagen nur eine implizite Verwendung vor, d.h. nur die Nutzung der ohnehin stattfindenden intuitiven Augenbewegungen.

Diese Arbeit untersucht die Steuerung von Computern mit komplexen Blickgesten und ist wie folgt aufgebaut: In der ersten Nutzerstudie wird verifiziert, dass Menschen durchaus in der Lage sind, willentlich komplexe Blickgesten durchzuführen. Zusätzlich werden die natürlichen Augenbewegungen nach dem Auftreten von Blickgesten untersucht. In einer weiteren Phase werden mögliche Einsatzgebiete für die Blickgesten experimentell evaluiert. In der zweiten Benutzerstudie werden die Parameter des Erkennungsalgorithmus optimiert, so dass die Blickgesten sicher von natürlichen Augenbewegungen unterschieden werden können.

## 2 Blickgesten

Gesten zur Eingabe sind ein bekanntes Konzept und wurden vielfach untersucht. Typische Beispiele, die sich auf manuelle Eingabe beziehen, sind Unistroke (Goldberg & Richardson 1993) und Cirrin (Mankof & Abowd 1998). Erstaunlicherweise wurden Gesten, die mit den Augen vollführt werden, bisher kaum studiert. Qvarfordt und Zhai (Qvarfordt & Zhai 2005)

stellten ein Gesten-basiertes Dialogsystem vor. Die verwendeten Gesten sind jedoch die intuitiven Augenbewegungen und nicht Gesten, die erst gelernt und dann reproduziert werden müssen, was aber den Einsatzbereich stark limitiert. Explizite Gesten werden von Isokoski (Isokoski 2000) eingeführt. Zur Eingabe von Zeichen werden Ziele außerhalb des Bildschirms in bestimmter Reihenfolge mit dem Blick angesteuert. Diese Gesten sind jedoch fest in der Größe und an fixe Positionen gebunden. Diese Arbeit führt das Konzept weiter und eliminiert die Probleme, welche durch Blickziele außerhalb des Bildschirms auftreten.

Die hier vorgestellten Blickgesten basieren nur auf Relativbewegungen und sind darum völlig immun gegen Kalibrierungsdrift. Die Anforderungen an die Genauigkeit sind um den Faktor 5 bis 15 geringer als die Genauigkeit des für die Studie verwendeten Blickverfolgers. Durch die geringen Anforderungen kann eventuell auf eine Kalibrierung verzichtet werden. Da die Genauigkeit von Blickverfolgern eine Winkelgröße ist, erlauben geringe Genauigkeitsanforderungen auch größere Entfernungen zwischen Auge und zu bedienendem Gerät.

## 2.1 Die Grundlagen des Gestenerkennungsalgorithmus

In der ersten Phase dieser Forschung stand die Frage im Vordergrund, ob Personen überhaupt in der Lage sind, komplexe Blickgesten durchzuführen. Die Wahl der Gesten war zu diesem Zeitpunkt zweitrangig. Da die Mausgesten, die es als Plug-in<sup>3</sup> für den Firefox-Webbrowser gibt, sich großer Beliebtheit erfreuen, wurde mit diesen Gesten begonnen.

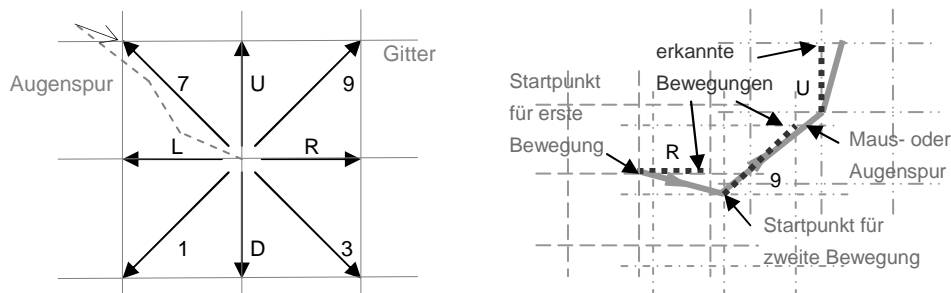


Abbildung 1: Bewegungen werden auf Gitterpunkte abgebildet und in acht Richtungen übersetzt.

Der Gestenalgorithmus verwendet ein Gitter mit Abstand  $s$ . Jede Position einer Maus- oder Augenspur kann durch einfache Integer-Division einem Gitterpunkt zugeordnet werden. Sobald eine Bewegung bezüglich ihres Startpunkts eine Position erreicht, die einem neuen Gitterpunkt entspricht, wird die Bewegung in eine der acht Richtungen übersetzt (siehe Abb. 1). Die Richtungen werden mit je einem der Buchstaben U, D, L, und R oder einer der Ziffern 1, 3, 7, und 9, entsprechend der Anordnung der Ziffern auf dem Ziffernblock, ausgedrückt. Eine Geste ist eine Folge von Bewegungen, die durch eine Zeichenkette beschrieben

<sup>3</sup> <http://optimoz.mozdev.org/gestures>

werden kann. Die Gestenerkennung ist folglich ein zweistufiger Prozess. Im ersten Schritt werden die Bewegungen in eine Zeichenkette übersetzt und im zweiten Schritt wird in dieser Zeichenkette nach einem Muster gesucht, welches die Geste beschreibt.

Ein weiterer Algorithmus, der diese Arbeit inspirierte, ist EdgeWrite (Wobbrock et al. 2003). Dieser Algorithmus verwendet die vier Ecken eines Quadrats und eine Geste setzt sich aus Bewegungen entlang der sechs Verbindungslinien zusammen. Wie man leicht sieht, können alle EdgeWrite-Gesten mit den Richtungsnamen der Mausgesten beschrieben werden und bilden damit eine Untermenge der oben beschriebenen Mausgesten. Dies ist interessant, weil die Erfinder der EdgeWrite-Gesten zeigten, dass mit ihren Gesten ein umfangreiches Alphabet beschrieben werden kann.

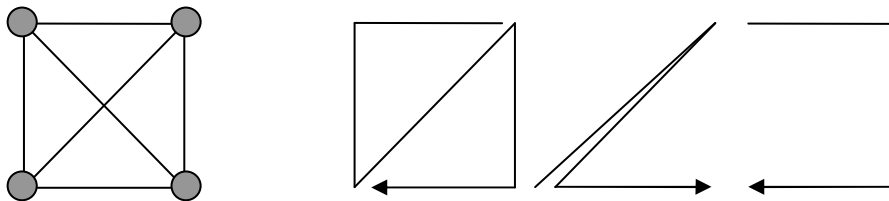


Abbildung 2: Die EdgeWrite-Gesten setzen sich aus Bewegungen entlang der sechs Verbindungslinien der vier Ecken eines Quadrats zusammen. Die drei Beispielgesten stehen für die Ziffern 0, 2 und 3.

## 2.2 Die Implementierung des Gestenerkennungsalgorithmus

Der als Vorbild dienende Mausgestenalgorithmus empfängt bei gedrückter Gestentaste aktuelle Koordinaten und bildet die Differenz zu einem Startpunkt. Falls weder die  $x$ - noch die  $y$ -Komponente der Differenz die Gittergröße  $s$  überschreitet, wird auf die nächste Koordinate gewartet. Andernfalls wird die Differenz in eines der acht Zeichen entsprechend der Bewegungsrichtung ausgegeben (siehe Abb. 1). Die aktuellen Koordinaten werden zum Startpunkt der nächsten Bewegung. Eine Geste (siehe Abb. 3) wird erkannt, wenn die korrespondierende Zeichenkette ausgegeben wurde.

Die Mausgesten bestehen durch ihre Einfachheit, haben jedoch den Nachteil, eine Gestentaste, normalerweise die rechte Maustaste, zu verwenden. Da die Blickgesten ohne zusätzliche Eingaben auskommen sollen, wurde der Blickgestenalgorithmus auf permanente Erkennung, also ohne zusätzlichen Tastendruck, umgestellt. Hierzu müssen sich die zu erkennenden Gesten von den natürlich durchgeführten Augenbewegungen unterscheiden.

Um besser zwischen natürlichen Augenbewegungen und willentlich ausgeführten Blickgesten unterscheiden zu können, wurde ein Zeitkriterium (Timeout) eingeführt. Wenn für eine bestimmte Zeit  $t$  keine Bewegung erkannt wurde, gibt der Blickgestenalgorithmus als neuntes mögliches Zeichen einen Doppelpunkt aus. Jede längere Fixation setzt dadurch die Gestenerkennung zurück. Während der Durchführung einer Blickgeste sollten nur kurze Fixationen auftreten.

## 3 Experimente und Benutzerstudien

Alle Benutzerstudien wurden auf einem ERICA<sup>4</sup> Blickverfolger durchgeführt. Dieser Blickverfolger läuft auf einem Tablet-PC mit 1,4 GHz Prozessorfrequenz unter dem Betriebssystem Windows XP. Die Bildschirmauflösung beträgt 1024 x 768 Pixel. Da das System keinen Kopfverfolger besitzt, mussten die Versuchspersonen ihren Kopf auf einer Kinnstütze ruhig halten. Der Abstand der Augen zum Bildschirm betrug 45 cm, womit 1° Blickwinkel 36 Pixel auf dem Bildschirm entspricht. Die Genauigkeit ist mit  $\pm 0,5^\circ$  angegeben. Der Blickverfolger stellt eine Programmierschnittstelle zur Verfügung, die während Fixationen x- und y-Koordinaten mit einer Rate von 60 Hz liefert. Während der Sakkaden werden keine Koordinaten geliefert. Die Software zur Erkennung der Blickgesten und zur Durchführung der Experimente und Benutzerstudien wurde im Forschungsprojekt selbst entwickelt.

### 3.1 Die erste Benutzerstudie

Die erste Benutzerstudie stand unter der Fragestellung, ob und bis zu welcher Komplexität die Versuchspersonen in der Lage sind Blickgesten auszuführen. Die Benutzerstudie wurde mit neun Personen, drei weiblich und sechs männlich, im Alter von 23 bis 47 Jahren, durchgeführt. Als Parameter des Gestenalgorithmus wurde für die Gittergröße  $s$  80 Pixel und für den Timeout  $t$  1000 ms gewählt. Es wurden drei Aufgaben gestellt.

Die erste Aufgabe bestand darin einen Dialog mit Ja oder Nein zu beantworten, indem hierzu die Ecken des Dialogs mit dem Blick im bzw. gegen den Uhrzeigersinn angeschaut wurden. Die Gestenerkennung detektierte die Gesten RDLU, DLUR, LURD und URDL als Ja und die Gesten DRUL, RULD, ULDR und LDRU als Nein. Es war somit egal, an welcher Ecke die Geste begonnen wurde. Alle Versuchspersonen waren sofort und mit Leichtigkeit in der Lage den Dialog mit einer Blickgeste zu beantworten. Tabelle 1 zeigt die gemessenen Zeiten.

Geste	Zeitaufwand für die Geste	Standardabweichung	Pro Segment
Ja (im Uhrzeigersinn)	1905 ms	613 ms	476 ± 153 ms
Nein (gegen den Uhrzeigersinn)	1818 ms	666 ms	455 ± 167 ms

Tabelle 1: Benötigte mittlere Zeit über alle Personen für die Ja- und Nein-Gesten.

Die zweite Aufgabe war die Durchführung von Blickgesten verschiedener Komplexität auf verschiedenen Hintergründen. Abbildung 3 zeigt die drei ausgewählten Gesten. Die drei Hintergründe waren die Anzeige von Hilfslinien, ein Bildschirmabzug einer Tabellenkalkulation und eine einfarbige graue Fläche. Jede der drei Gesten musste je dreimal auf drei verschiedenen Hintergründen ausgeführt werden. Alle Testpersonen waren in der Lage die drei Gesten bei Anzeige der Hilfslinien und bei Anzeige der Tabellenkalkulation durchzuführen. Auf einfarbigem Hintergrund konnte die schwerste Geste RD7DR7 noch von fünf der neun

<sup>4</sup> <http://www.eyeresponse.com/>

Personen innerhalb des Zeitlimits von drei Minuten durchgeführt werden. Die Zeiten zur Durchführung der Gesten sind in Tabelle 2 wiedergegeben. Aus den Zahlen wird ersichtlich, dass die Zeiten von der Anzahl der Gestensegmente, aber kaum vom Hintergrund abhängen.

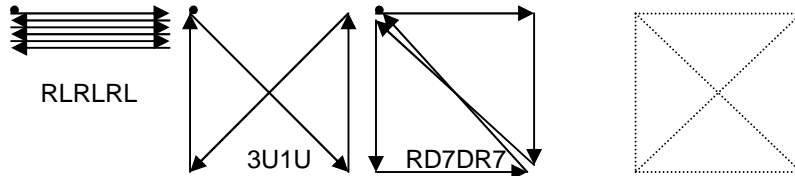


Abbildung 3: Die Abbildung zeigt drei in der Benutzerstudie verwendeten Blickgesten und die Hilfslinien.

Geste	Hintergrund		
	Hilfslinien	Tabellenkalkulation	einfarbig
RLRLRL	3106 ms	3136 ms	3379 ms
3U1U	2219 ms	2208 ms	2376 ms
RD7DR7	3153 ms	3671 ms	3588 ms

Tabelle 2: Zeiten für die Durchführung der verschiedenen Blickgesten auf verschiedenen Hintergründen.

Die dritte Aufgabe bestand darin für circa drei Minuten im Internet zu surfen. Währenddessen lief die Gestenerkennung und protokollierte die resultierenden Gesten in eine Datei. Hiermit sollte untersucht werden, welche Gesten während normaler Interaktion mit dem Computer auftreten. Insgesamt ergab sich eine Gestenzeichenkette aus 2737 Zeichen bei einer Gesamtzeit von 1700 Sekunden. Die RLRLRL-Geste trat sehr oft auf, da diese Geste der natürlichen Augenbewegung beim Lesen entspricht. Die Gesten 3U1U und RD7DR7 traten kein einziges Mal auf und eignen sich somit als Geste für allgemeine Steuerung.

## 3.2 Experimente

Nach den sehr ermutigenden Ergebnissen der Benutzerstudie stellte sich die Frage nach Einsatzgebieten der Blickgesten. Die EdgeWrite-Gesten definieren zwar ein volles Alphabet, aber die Blickgesten eignen sich nicht sonderlich für Texteingaben, da die Eingabe eines Zeichens zu lange dauert. Deswegen wurde die Idee verfolgt, mit Blickgesten Kommandos auszuführen. Um hierzu Experimente durchzuführen wurde das Testprogramm um eine Liste von Gesten mit zugeordneten Kommandos erweitert. Als Kommandos wurden die WM\_APPCOMMAND-Nachrichten des Windows-Betriebssystems implementiert, die es ermöglichen, Dokumente zu öffnen, zu speichern und zu schließen, im Webbrowser zu navigieren oder Medienanwendungen zu steuern.

Das Speichern von Dokumenten durch Blickgeste fand wenig Anklang. Das Hauptargument dagegen ist das Tastenkürzel Strg+S, welches dieselbe Funktion schneller ausführt. Die Medienkommandos erwiesen sich jedoch als interessant, da Medien oft aus einem Abstand zum Bildschirm konsumiert werden. Mediengeräte werden typischerweise mit Fernbedienungen

gesteuert. Blickgesten funktionieren wegen ihrer geringen Genauigkeitsanforderungen auch aus größerer Entfernung und die Ränder des Bildschirms ergeben natürliche Hilfslinien. Die Kamera des verwendeten Blickverfolgers ließ eine Fokussierung bis zu einem Abstand von einem Meter zu. Alle Testpersonen waren in der Lage auf diesem Abstand den Media Player mit Blickgesten zu steuern. Die zusätzlich eingeblendeten Hilfslinien erwiesen sich als unnötig. Die Testpersonen bevorzugten große Gesten und führten ihren Blick lieber entlang der Kanten des Videobilds. Dabei stellte sich heraus, dass die Blickgesten nicht quadratisch sein müssen, sondern mit allen Seitenverhältnissen gut funktionieren (siehe Abb. 4).

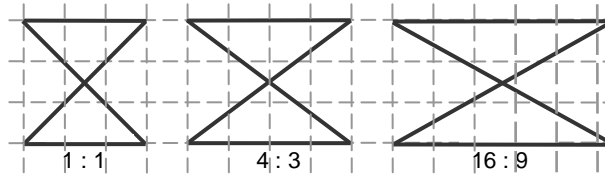


Abbildung 4: Die Blickgesten sind unabhängig vom Seitenverhältnis.

### 3.3 Die zweite Benutzerstudie

Die zweite Benutzerstudie wurde mit dem Ziel durchgeführt, die Parameter des Gestenalgorithmus dahingehend zu optimieren, dass Verwechslungen mit natürlichen Augenbewegungen ausgeschlossen werden können. Hierzu wurde der Timeout verringert. Aus der ersten Benutzerstudie hatte sich eine Zeit von 560 ms pro Gestensegment mit einer Standardabweichung von etwa 150 ms ergeben. Der Timeout wurde darum von 1000 ms auf 700 ms reduziert. Des Weiteren wurde die Gittergröße von 80 auf 250 Pixel erhöht. Die Motivation hierfür besteht in der Tatsache, dass Sakkaden über lange Distanzen relativ selten auftreten. Es ist vernünftig anzunehmen, dass der Blick sich meist innerhalb des Bildschirms bewegt und Blickgesten entlang der Bildschirmränder sich klar von natürlichen Augenbewegungen unterscheiden lassen. Die Testsoftware wurde ebenfalls modifiziert. Statt nur erkannte Gestensegmente zu protokollieren, wurden jetzt die Augenbewegungen protokolliert. Dies ermöglichte die aufgezeichnete Augenspur mit verschiedenen Parametern in Gesten umzurechnen.

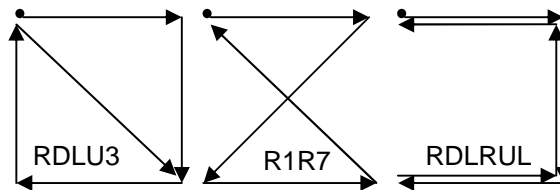


Abbildung 5: Die drei Blickgesten der zweiten Nutzerstudie.

Bei gleichem Ablauf wie in der ersten Benutzerstudie wurden nur die drei Gesten durch andere ersetzt (siehe Abb. 5), um Daten für ein größeres Repertoire möglicher Gesten zu

erhalten. Zusätzlich zu den drei Minuten surfen im Internet, wurden auch noch die Augenbewegungen beim Betrachten eines knapp vierminütigen Videos aufgezeichnet. An der Nutzerstudie nahmen 9 Personen, eine davon weiblich, im Alter von 11 bis 50 Jahren teil. Zwei Personen hatten bereits an der ersten Studie teilgenommen. Sieben der neun Testpersonen waren in der Lage die drei Gesten auf allen Hintergründen durchzuführen. Die anderen beiden Personen konnten immerhin sechs bzw. sieben der neun Gestenkombinationen ausführen. Erstaunlicherweise scheiterte eine Personen bei angezeigten Hilfslinien, war aber anschließend in der Lage die Geste auf leerem Hintergrund auszuführen. Die Daten des elfjährigen Kinds zeigen keine Auffälligkeiten. Für eine Person in der Studie erwies sich der verkürzte Timeout als schwierig. Die Analyse der aufgezeichneten Augenspuren ergab, dass diese Person bei einem Timeout von 1000 ms die Gesten auf Anhieb hätte erfüllen können, während sie mit 700 ms Timeout oft über eine Minute benötigte.

$s=80$ $t=1000$	:3LUD::7R1L9:73LR:73LR:7379RL:U:D:U3:LU::RL::R13U::LR:R:73:73D:7 3:LRLRLR7373DU7LD:RUL13L:R:RL:RL:LRL:R7L3L9R1UR3DR::7:URLR
$s=80$ $t=700$	:3LU:D::7R:1L9:7:3LR::73L:R::737:9:RL:U:D:U3:LU::RL::R13U::LR:R:73:7 3D:73:LR:LRLR7373DU7LD:RUL1:3L:R:RL:RL:LRL::R:7L:3:L9R:1UR3DR
$s=250$ $t=1000$	:3L::7R:1U:U3::7RL:UDU:L::::RD:R::::73::73::::L::L::R::L:R:L::R7:R:RL :DR:L::::U:R:R::LR:L:RL:UD:R::::L::LR:L:3:L::::D:RL:RLRL::DRL::RLRL
$s=250$ $t=700$	:3L::7R:1:U:U:3::7RL::UDU:L::::RD:R::::73::73::::L::L::R::L:R:L: ::R7::R::R:L:D:R:L::::U:R:R::LR::L:RL:UD:R::::L::LR:L:3:L::::D::

Tabelle 3: Gestenberechnung für dieselbe Augenspur (Surfen im Internet) mit jeweils verschiedenen Parametern. Die erste Zeile entspricht den Parametern der ersten Studie, die letzte Zeile den Parametern der zweiten Studie.

Geste	Anzahl aufgetretener Gesten während surfen im Internet (1/2 h)				Anzahl aufgetretener Gesten während Video anschauen (1/2 h)			
	$s=80$	$s=80$	$s=250$	$s=250$	$s=80$	$s=80$	$s=250$	$s=250$
	$t=1000$	$t=700$	$t=1000$	$t=700$	$t=1000$	$t=700$	$t=1000$	$t=700$
RDLU	3	2	0	0	0	0	0	0
DLUR	4	1	0	0	1	1	0	0
LURD	1	0	1	0	0	0	0	0
URDL	3	2	0	0	1	1	0	0
DRUL	6	4	0	0	1	0	0	0
RULD	3	1	0	0	0	0	0	0
ULDR	2	0	0	0	0	0	0	0
LDRU	1	1	0	0	0	0	0	0
RLRLRL	41	22	5	0	7	3	5	3
3U1U	0	0	0	0	0	0	0	0
RD7DR7	0	0	0	0	0	0	0	0
RDLU3	0	0	0	0	0	0	0	0
R1R7	2	1	0	0	0	0	0	0
RDLRUL	1	1	0	0	0	0	0	0
# char	3.447	4.022	2.651	3.391	3.182	3.690	2.326	2.873
char / s	1,62	1,89	1,25	1,59	1,59	1,85	1,16	1,44

Tabelle 4: Gestenaufkommen bei Berechnung mit verschiedenen Parametern.



Die Umrechnung der aufgezeichneten Augenspuren beim Surfen (siehe Tabelle 3) und Video anschauen hat ein interessantes Ergebnis. Die geänderten Parameter reduzieren die Wahrscheinlichkeit, in den natürlichen Augenbewegungen eine Geste zu erkennen, auf nahezu null. Schon die Änderung der Gittergröße genügt zur Erreichung dieses Ziels. Mit Ausnahme der RLRLRL-Gesten wurden sowohl im den aufgezeichneten 35 Minuten Augenbewegungen beim Surfen, als auch bei den 33 Minuten Video anschauen keine der Gesten aus den Nutzerstudien gefunden. Dies gilt für die Umrechnung mit  $t = 700$  ms und  $s = 250$  Pixel und auch für  $t = 1000$  ms und  $s = 250$  Pixel (siehe Tabelle 4).

## 4 Anwendungen für Blickgesten

Die Blickgesten erwiesen sich als ernsthafte Alternative zur bisher verwendeten Augensteuerung, die auf der Dwell-Time-Methode basiert. Die Blickgesten lösen die Probleme mit Kalibrierung und Genauigkeit, die mit der Dwell-Time-Methode auftreten.

Für die Blickgesten bieten sich zwei Verwendungen an. Die erste Verwendung betrifft diejenigen, die auf Blickverfolgersysteme angewiesen sind. Wegen der Unempfindlichkeit gegenüber Kalibrierungsfehlern kann eine Blickgeste dazu verwendet werden einen Rekalibrierungsvorgang zu starten. Auch die Verwendung von Blickgesten für Kommandos, die normalerweise über Tastenkürzel angesprochen werden, ist denkbar.

Eine andere Verwendung von Blickgesten ist die Fernsteuerung von Mediengeräten als Alternative zu Fernbedienung. Da Gesten unabhängig von der absoluten Position sind, erscheint es möglich mit einer einmaligen Ausrichtung der Kamera und ohne weitere Kalibrierungen die Gestenerkennung zu erreichen. Die Kanten des Bildschirms bilden dann die Hilfslinien als Orientierung für die Blickgeste. In (Vertegaal et al. 2005) wird vorgeschlagen eine grobe Erkennung des Blicks (z.B. welches Gerät wird betrachtet) zu nutzen um die Fernbedienungen mehrerer Geräte durch eine einzige Fernbedienung zu ersetzen. Diese steuert dann das Gerät, auf das der Blick des Benutzers gerichtet ist. Mit der Blickgesten-Steuerung kann auch noch die letzte Fernbedienung entfallen.

In den durchgeführten Studien erwiesen sich die Blickgesten als ein interessantes und neuartiges Konzept für die Computer-Interaktion. Weitere Forschung auf dem Gebiet, z.B. alternative Blickgesten oder Erlernbarkeit eines Blickgesten-Vokabulars, erscheinen lohnenswert.

### Literaturverzeichnis

- Bolt, R. A.; (1981): Gaze-orchestrated dynamic windows. In: Proceedings of the 8th Annual Conference on Computer Graphics and interactive Techniques SIGGRAPH '81. S. 109-119
- Goldberg, D.; Richardson, C. (1993): Touch-typing with a stylus. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '93. S. 80 – 87
- Isokoski, P. (2000): Text input methods for eye trackers using off-screen targets. In: Proceedings of the 2000 Symposium on Eye Tracking Research & Applications ETRA '00. S. 15-21

- Jacob, R. J. (1990): What you look at is what you get: eye movement-based interaction techniques. In: Chew J. C.; Whiteside J. (Hrsg.): Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '90. S. 11-18.
- Majaranta, P.; Rähkä, K. (2002): Twenty years of eye typing: systems and design issues. In: Proceedings of the 2002 Symposium on Eye Tracking Research & Applications ETRA '02. S. 15-22.
- Mankof, J.; Abowd, G. D. (1998): Cirrin: a word-level unistroke keyboard for pen input. In: Proceedings of the 11th Annual ACM Symposium on User interface Software and Technology UIST '98. S. 213-214.
- Qvarfordt, P.; Zhai, S. (2005): Conversing with the user based on eye-gaze patterns. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '05. S. 221-230.
- Salvucci, D. D.; Anderson, J. R. (2000): Intelligent gaze-added interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '00. S. 273-280.
- Špakov, O.; Miniotos, D.: Gaze-based selection of standard-size menu items. In: Proceedings of the 7th international Conference on Multimodal interfaces ICMI '05. S. 124-128.
- Vertegaal, R.; Mamuji, A.; Sohn, C.; Cheng, D.: (2005): Media Eyepliances: Using Eye Tracking for Remote Control Focus Selection of Appliances. In: Extended Abstracts on Human Factors in Computing Systems CHI '05. S. 1861-1864.
- Ware, C.; Mikaelian, H. H. (1987): An evaluation of an eye tracker as a device for computer input. In: Carroll, J. M.; Tanner, P. P. (Hrsg.): Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics interface CHI '87. S. 183-188.
- Wobbrock, J. O.; Myers, B. A.; Kembel, J. A. (2003): EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In: Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology UIST '03. S. 61-70.
- Zhai, S.; Morimoto, C.; Ihde, S.: Manual and Gaze Input Cascaded (MAGIC) Pointing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '99. S. 246-253.

### **Danksagung**

Die hier beschriebene Forschungsarbeit wurde in Teilen im Rahmen der Nachwuchsgruppe „Eingebettete Interaktion“ im Aktionsplan Informatik von der DFG gefördert.

### **Kontaktinformationen**

Nachwuchsgruppe „Eingebettete Interaktion“ und  
Lehr- und Forschungseinheit Medieninformatik  
Institut für Informatik  
Ludwig-Maximilians-Universität München  
Heiko Drewes, Heinrich Hußmann und Albrecht Schmidt

Amalienstraße 17  
80333 München  
heiko@hcilab.org, heinrich.hussmann@ifi.lmu.de, albrecht.schmidt@hcilab.org

Tel.: +49 89/2180-4650