

Fachseminar - Verteilte Systeme

Alternative Texteingabe für Mobilgeräte

Patrick Jayet - jayetp@student.ethz.ch

7. Juni 2005

Motivation

Das Thema der alternativen Texteingabe für Mobilgeräte ist wegen verschiedenen Faktoren wichtig. Einerseits nimmt die Miniaturisierung der Geräte im allgemeinen zu. Es geht in die Richtung des Ubiquitous Computing. Andererseits nimmt die Rechenleistung der Geräte auch zu. Es ist heute üblich, einen 100 bis 200 Mhz Prozessor in einem Smartphone zu finden. Dies ermöglicht insbesondere, neue Anwendungen auf diesen Mobilgeräten laufen zu lassen. Neben klassischen Killer-Applikationen wie SMS gibt es immer mehr Bedarf, E-Mail Applikationen sowie Agenda und Wortprozessoren auf diesen Geräten zu verwenden.

Eine klassische PC-Tastatur kann wegen Platzbeschränkung nicht integriert werden. Die Entwicklung und der Einsatz von alternativer Texteingabe bleibt also wichtig.

Ziel

Die Ziele dieser Arbeit sind einerseits ein Überblick über den Bereich der Texteingabe zu gewinnen, insbesondere seine Problematiken und die verschiedene existierende Technologien kennenzulernen, andererseits ein Einblick in zwei aktuelle Forschungsprojekte zu erhalten.

Übersicht

In diesem Seminar werden in der Einführung Themen wie die Problematik bei der Bewertung eines Eingabeverfahrens sowie verschiedene Eingabeverfahren aus den Kategorien tastaturbasiert, stylusbasiert und unkonventionelle Eingabe behandelt. Danach werden die 2 aktuellen Forschungsprojekte Dasher und Presense im Detail angeschaut.

1 Einführung

1.1 Problematik bei der Bewertung eines Eingabeverfahrens

1.1.1 Texterzeugung vs Textkopie

Das Ziel der Bewertung eines Verfahren ist eine Aussage über die Leistung des Verfahrens unter realistischen Bedingungen zu formulieren. Es entspricht einem Experiment, in dem der Tester Texterzeugung durchführt. Es gibt aber damit ein paar Probleme. Insbesondere gibt es keine Kontrolle über die Verteilung der Buchstaben und der Texterzeugungsprozess (einen Text ausdenken) kann Zeit kosten, was das Resultat eventuell verschlechtern kann. Deswegen wird in der Praxis die Textkopiemethode verwendet. Diese Methode hat aber auch Nachteile. Zusätzlich zur Bedienung des Texteingabesystemes muss der Tester auch den Text, den er abschreibt, lesen. Es erzeugt also einen sogenannten *Focus of Attention* (FOA) mehr.

Bsp. PC-Tastatur Für eine PC-Tastatur und einen fortgeschrittener Benutzer entspricht die Textkopiermethode einen Focus of Attention, der Benutzer braucht nur die Kopie zu lesen. Für ein Anfänger gibt es 1 FOA zusätzlich: der Tester muss auf den Bildschirm schauen, um mögliche Fehler zu erkennen und zu korrigieren. Insgesamt also 2 FOA.

Bsp. Softtastatur oder stylusbasierte Eingabe Eine Softtastatur¹ oder eine stylusbasierte Eingabe kann nicht ohne Aufmerksamkeit bedient werden. Man muss den Bildschirm oder den Stylus anschauen. Deswegen gibt es 1 Focus of Attention mehr als für eine PC-Tastatur. Insgesamt also 2 FOA für einen vorgeschrittenen Benutzer und 3 FOA für einen Anfänger.

1.1.2 Anfänger vs Fortgeschrittene

Das Ziel eines guten Systems ist einerseits, dass ein Anfänger relativ rasch eine akzeptable Leistung erzielt. Ansonsten wird das System nie tatsächlich verwendet, weil nur wenige Benutzer genug Zeit dafür investieren. Andererseits möchte man, dass fortgeschrittene Benutzer eine gute Leistung erreichen können.

Die Leistung wird durch das Potenz-Gesetz des Lernens modelliert, eine empirische Beziehung, die das Lernverhalten bei der Wiederholung von Handlungen numerisch modelliert. Die Beziehung hat die folgende Form:

$$P(n) = A + B * n^\alpha$$

wobei n die Anzahl Wiederholungen darstellt und A, B, α Konstanten sind.

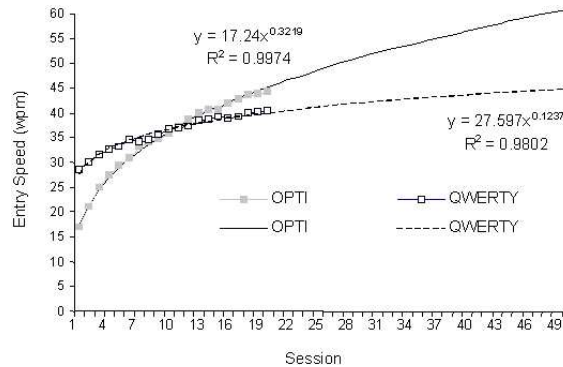


Abbildung 1: Lernkurven zweier Verfahren modelliert durch das Potenz-Gesetz des Lernens.

Das Potenz-Gesetz des Lernens ermöglicht zugleich verschiedene Verfahren miteinander zu vergleichen. Man sieht zum Beispiel in der Figur 1 die Kurve zweier verschiedenen Verfahren. Obwohl das erste für einen Anfänger schlechter als das zweite ist, wird es für einen fortgeschrittenen Benutzer umgekehrt. Der Vergleich zwischen verschiedenen Verfahren wird am besten durch eine systematische Studie und eine Modellierung mit dem Potenz-Gesetz des Lernens geführt.

1.1.3 Einheit der Geschwindigkeit

Die Leistung von Verfahren wird entweder in *Zeichen pro Sekunde* (cps) oder in *Wörter pro Minute* (wpm) gemessen. In dieser Arbeit wird die Leistung in Wörter pro Minute ausgedrückt. Bei dieser Einheit besteht ein Wort aus 5 Zeichen unabhängig davon, ob Leerzeichen oder nicht. Eine einfache Beziehung zwischen den zwei Beziehungen ist deshalb gegeben durch:

$$G_{wpm} = G_{cps} \frac{60}{5} = 12G_{cps}$$

¹Eine Softtastatur ist eine Tastatur, die in Software realisiert wird, das heisst, dass eine Tastatur auf dem Bildschirm angezeigt wird, auf welcher man mit dem Maus einen Buchstaben wählen kann.

1.2 Tastaturbasierte Eingabe

1.2.1 Mehrdeutigkeit der Tasten

Für tastaturbasierte Texteingabe kann die Mehrdeutigkeit der Tasten als ein Kontinuum betrachtet werden. Als Extremfall hat man eine Taste für jedes Zeichen, unabhängig ob gross oder klein. Ein anderer Extremfall wäre, dass alle Zeichen auf derselben Tasten abgebildet werden. Zwischen diesen 2 Extremfällen gibt es verschiedene interessante Anhaltspunkte. In dieser Arbeit werden insbesondere eine Variante der Qwerty-Tastatur sowie die Tastatur des Mobiltelefons betrachtet.

1.2.2 Anhaltspunkt 1: Tastatur eines Mobiltelefons

Bei dieser Tastatur werden die Tasten 2-9 zugleich für Buchstaben verwendet. Die Mehrdeutigkeit, die in irgendeiner Weise aufgelöst werden muss, beträgt 3 bis 4 Zeichen pro Taste. Dafür gibt es verschiedene Ansätze. In diesem Artikel werden wir insbesondere Multitap, T9 und TiltText in Betracht nehmen.

Multitap Multitap ist die meistbenutzte Methode, um die Mehrdeutigkeit einer Telefonstastatur aufzulösen. Ein mögliches Problem dabei besteht darin, die Aufteilung einer Sequenz zu bestimmen. Wenn man zum Beispiel die Kette ON schreiben will, muss man 5 mal auf die Taste 6 drücken (3 mal für O und 2 mal für N). Es muss also eine Methode geben, um die Trennung der 2 Buchstaben zu bestimmen. Dies wird durch ein Timeout oder eine Timeout-Kill Taste erreicht. Üblicherweise beträgt ein Timeout auf einem Nokia Mobiltelefon etwa 1.5 Sekunde. Die Leistung von Multitap beträgt für einen fortgeschrittenen Benutzer 20 bis 25 wpm.

T9 Bei der T9 Methode wird ein Wörterbuch zur Auflösung der Mehrdeutigkeit verwendet. Die Leistung für einen fortgeschrittenen Benutzer beträgt 40 bis 45 wpm. Dies wird allerdings nur unter bestimmten optimistischen Bedingungen erreicht. Es wird angenommen, dass alle Wörter sich im Wörterbuch befinden, sowie dass keine Mehrdeutigkeit für die Wörter besteht, das heisst, dass das gewünschte Wort immer als erstes in der Liste steht — das Drücken der **next** Taste ist also nicht nötig. Ausserdem nimmt die Leistung schnell ab, falls spezielle Wörter wie Slang, Eigennamen oder Abkürzungen verwendet werden, was sehr oft der Fall ist in der Praxis — insbesondere für SMS. Unter diesen Umständen kann die angegebene Leistung nur als idealisierte obere Grenze betrachtet werden. Als Vergleichspunkt kann man die Leistung für eine Computertastatur erwähnen. Sie beträgt 50 bis 70 wpm für einen fortgeschrittenen Benutzer.

TiltText Bei dieser Methode wird die Mehrdeutigkeit der Tasten durch das Kippen des Mobiltelefons aufgelöst. Die verschiedene Kippbewegung sind in Abbildung 2(a) dargestellt. Das Mobiltelefon kann in die 4 Richtungen gekippt werden. Man sieht in Abbildung 2(b) wie die Mehrdeutigkeit der Taste 7 aufgelöst wird. Das Kippen nach links wählt den ersten Buchstaben P, nach vorne den zweiten Buchstaben Q, nach rechts den Dritten und nach hinten den Vierten. Das Drücken der Taste 7, ohne das Mobiltelefon zu kippen, wählt die numerische Wert 7. Im Paper [Wigd03] wird behauptet, dass TiltText eine Leistung von etwa 20% besser als Multitap aufweist. Die benützte Implementierung von Multitap ist allerdings seltsam, weil das Timeout 2 Sekunden beträgt — verglichen mit etwa 1.5 Sekunden. Die angegebene Leistungen von TiltText und Multitap betragen dabei 13.5 bzw. 11 wpm. Ausserdem ist ein Nachteil von TiltText eine Fehlerrate, die 2 bis 3 mal höher ist als für Multitap.

1.2.3 Anhaltspunkt 2: Halb-Qwerty Tastatur

Der zweite Anhaltspunkt in diesem Kontinuum der Tastenmehrdeutigkeit ist die Halb-Qwerty Tastatur. Es geht dabei um eine Computertastatur, deren beide Hälften nur mit einer Hand bedient werden können. Es kann gewählt werden, in welcher Hälfte der Tastatur man einen Buchstabe tippen will, indem man gleichzeitig zur Taste die Leertaste gedrückt hält. Die Leistung dieses Systems beträgt nach MacKenzie et al. [MacKen02] 41-73% der Leistung einer Computertastatur nach 10 Stunden Praxis.

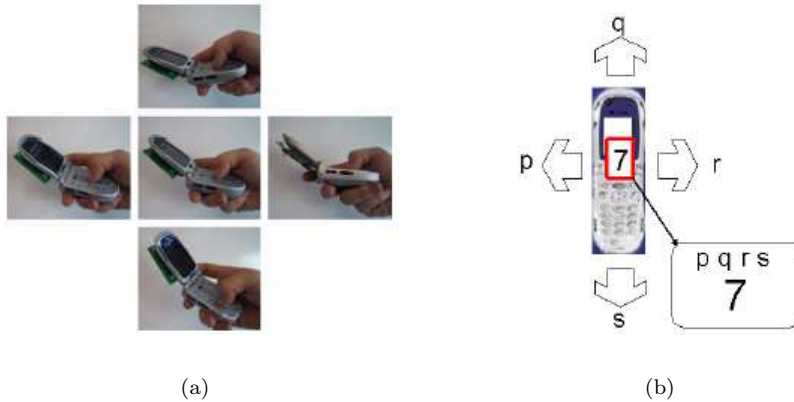


Abbildung 2: TiltText (a) Mögliche Kippbewegungen des Mobiltelefons (b) Möglichkeit ein Zeichen einer Taste durch das Kippen des Mobiltelefons in die verschiedene Richtungen einzugeben.

1.3 Stylusbasierte Eingabe

Bei stylusbasierten Texteingabe geht es darum, Text mittels Erkennung der Stylusbewegung einzugeben. Ein Nachteil vom Verfahren ist, dass beide Hände benötigt werden müssen — verglichen zum Beispiel mit der Tastatur vom Mobiltelefon. Die 2 Hauptprobleme, die für stylusbasierte Eingabe auftreten, sind einerseits die Segmentierung oder Trennung zwischen den verschiedenen Buchstaben und andererseits die Erkennung der Buchstaben selbst. Verschiedene Ansätze existieren, von denen wir kurz 2 klassische Beispiele betrachten werden, Graffiti und Unistrokes.

1.3.1 Graffiti

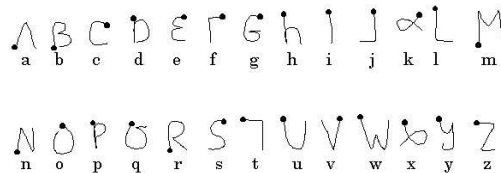


Abbildung 3: Graffiti Alphabet

Graffiti stellt eine leicht vereinfachte Version des Alphabets dar. Es behält die Ähnlichkeit mit dem Alphabet, was die Lernbarkeit des Verfahrens erleichtert. Das Problem der Segmentierung wird dadurch gelöst, dass jeder Buchstabe aus einem einzigen Strich besteht. Graffiti ist sehr wahrscheinlich ein signifikanter Erfolgsfaktor bei der Verbreitung vom Palm PDA. Die Leistung, die man für Graffiti erwarten kann, beträgt etwa 20 wpm für einen fortgeschrittenen Benutzer und 7 wpm für einen Anfänger.

1.3.2 Unistrokes

Unistrokes wurde 1993 im Xerox PARC von Goldberg und Richardson erfunden und entwickelt. Das Verfahren besteht aus einem reinen synthetischen Ansatz, weil es keine Ähnlichkeit mit dem Alphabet aufweist. Es macht daher die Lernbarkeit relativ schwierig. Keine systematische Studie von Unistrokes wurde gemacht, es wird nur erwähnt, dass eine Leistung von 34 wpm beobachtet wurde, was in dem Fall nicht sehr aussagekräftig ist. Das Verfahren wurde nie relativ breit verwendet und hat keinen Erfolg gehabt.

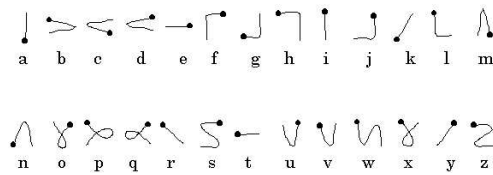


Abbildung 4: Unistrokes Alphabet

1.4 Unkonventionelle Eingabe

In dieser Sektion werden zwei unkonventionelle Texteingaben betrachtet.

1.4.1 Lightglove

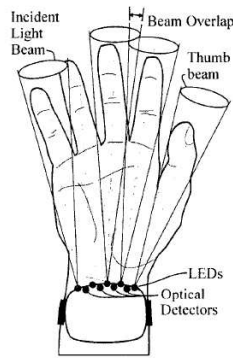


Abbildung 5: Schema des Lightglove

Das Lightglove System besteht aus einem kleinen Gerät, das unter dem Handgelenk platziert wird. Abstandssensoren, die durch IR Sensoren realisiert werden, ermöglichen die Fingerbewegung zu detektieren. Ausserdem gibt es Bewegungssensoren im Gerät, die die absolute Position des Lightglove im Raum bestimmen. Diese zwei Arten von Sensoren erlauben die verschiedenen Bewegungen der Hand und der Finger als Interaktion auf einer virtuellen Tastatur im Raum zu interpretieren. Für diese Methode wurde keine systematische Studie realisiert.

1.4.2 Thumbwheel

Das Thumbwheel System ist insbesondere geeignet, um Text in sehr kleinen Geräten einzugeben. Es besteht aus einem Drehrad, das eine X-Position auf einer Achse bestimmt, sowie aus einer Taste, die durch das Drücken auf dem Drehrad realisiert wird. Die Wahl eines Zeichens geschieht dadurch, dass man auf eine Liste geht und die Taste drückt, wenn der gewünschte Buchstabe erreicht wird. Zwei verschiedene Varianten wurden dabei getestet: einerseits eine einfache lineare Liste, andererseits eine hierarchische Liste, die in 2 Stufen strukturiert wird, d. h. man wählt zuerst eine Familie von Buchstaben, geht man dann in die zweite Stufe über, um die Selektion selbst durchzuführen.

Wie man es erwarten könnte, ist diese Methode bezüglich die Anzahl Tastendruck nicht besonders effizient. Im Mittel und je nach Variante — linear oder hierarchisch — werden pro Zeichen 8 bis 10 Tastendrucke² benötigt.

²Das Drehen des Drehrades von einer Position wird als Tastendruck interpretiert

2 Hybride Texteingabe: Dasher

2.1 Motivation

Bei der Qwerty-Tastatur entspricht eine Bewegung eine Entropie von etwa 6.3 Bits — 1 von 80 Tasten wird gedrückt. Die Entropie von englischen Texten wurde aber von Shannon auf 1 Bit pro Zeichen geschätzt [Shan93]. Bei einer PC-Tastatur werden also redundante Informationen eingegeben. Ausserdem ist die Rate der diskreten Bewegungsereignisse für einen Mensch begrenzt, was eine Limitation der Tippgeschwindigkeit darstellt.

Die Dasher Methode basiert auf einer Zeigeaufgabe (Pointing Task) mit kontinuierlicher Bewegung, deren Informationsbandbreite laut Drury und Hoffmann [Dru92] auf 14 Bits pro Sekunde geschätzt wurde. Dies würde theoretisch eine Leistung von 14 Zeichen pro Sekunde erlauben. Diese Annahme ist aber viel zu optimistisch und stimmt für Dasher nicht, was wir im folgendem sehen werden.

2.2 Einführung

Dasher gehört in der Kategorie der unkonventionellen Texteingabe. Das System basiert auf einer probabilistischen Sprachmodellierung und ist somit nicht Wörterbuch-basiert. Es wurde 1997 von MacKay et al. erfunden und bis heute kontinuierlich entwickelt. Das Ziel des Systems ist eine Texteingabe einerseits für Mobilgeräte und andererseits für bewegungsbehinderte Personen zu ermöglichen

2.3 Prinzip

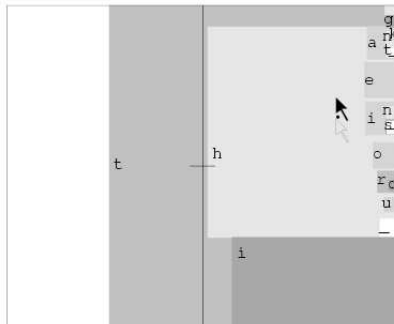


Abbildung 6: Screenshot von Dasher

Die Zeichen werden durch Flächen dargestellt (siehe Abbildung 6). Die Höhe der Fläche eines Buchstabens entspricht seiner Wahrscheinlichkeit in der Sprache gegeben die vorherige Zeichenkette. Man sieht z.B. in der Abbildung 6, dass die Höhe des Zeichens h gross ist, da im Englischen die Wahrscheinlichkeit von h nach t gross ist.

Das System wird durch eine kontinuierliche 2-dimensionale Bewegung gesteuert. Durch die X-Bewegung wird die Geschwindigkeit des Scrolling bestimmt, durch die Y-Bewegung wird ein Zeichen gewählt. Die Eingabe eines Zeichens erfolgt, wenn es die Mittellinie des Bildschirms überquert.

2.4 Höhe eines Buchstabens

Die Höhe eines Buchstabens wird durch die Formel von Bayes berechnet:

$$P(x_1 = a_1, x_2 = a_2)P(x_1 = a_1)P(x_2 = a_2|x_1 = a_1)$$

Für jeden Buchstaben wird eine minimale Wahrscheinlichkeit δ definiert (z.B. $\delta = 0.02$), damit seltene Buchstaben relativ einfach gefunden werden können.

2.5 Sprachmodellierung

Das benutzte Textmodellierungsverfahren heisst *Prediction by Partial Match* (PPM) und ermöglicht die Wahrscheinlichkeit eines Zeichens gegeben ein Kontext einer gewissen Länge zu berechnen. Es erlaubt englische Texte auf etwa 2 Bits pro Zeichen zu komprimieren. Obwohl dieses Verfahren nicht das wirksamste ist, wurde es gewählt wegen seiner Effizienz — die Wahrscheinlichkeit des nächsten Buchstabens muss bei jeder Eingabe neu berechnet werden.

2.6 Empirische Ergebnisse

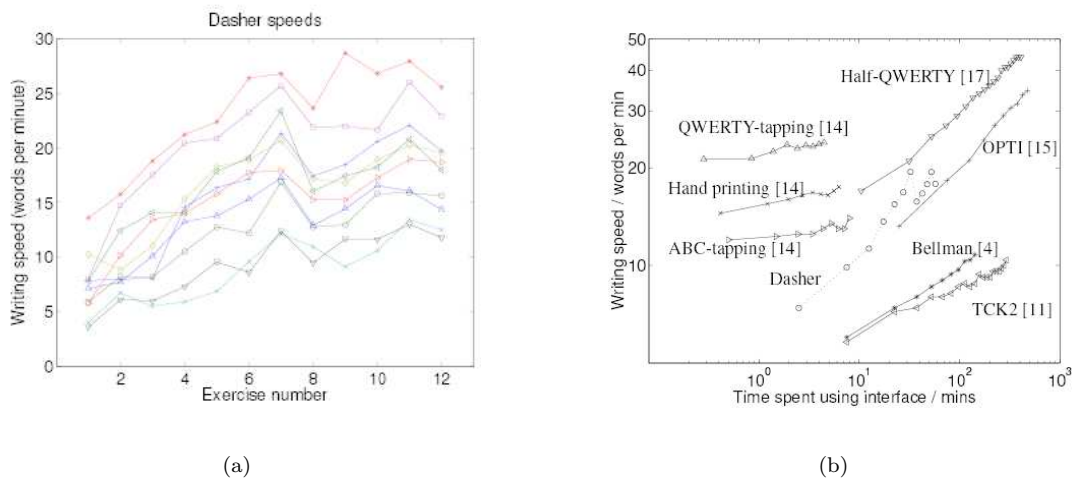


Abbildung 7: (a) Empirische Ergebnisse für die Leistung von Dasher in Funktion der Anzahl Übungssitzungen (b) Vergleich von Dasher mit anderen Techniken.

Eine empirische Studie wurde für Dasher durchgeführt, siehe Abbildung 7(a). Die Resultate wurden mit dem Potenz-Gesetz des Lernens, das wir in Sektion 1.1.2 gesehen haben, modelliert.

Man kann bei der Abbildung 7(a) sehen, dass ein fortgeschrittener Benutzer eine Leistung von etwa 20 wpm erreichen kann. Obwohl diese Leistung relativ gering bleibt — geringfügig kleiner als die Leistung von Multitap — ist das System gut brauchbar. Ausserdem kann Dasher selbst durch ein Augentrackingsystem mit der gleichen Leistung bedient werden. Die Fehlerrate bleibt relativ gering, unter 5% der Wörter enthalten Fehler (fortgeschrittener Benutzer).

2.7 Vergleich mit anderen Techniken

Man sieht auf der Abbildung 7(b), dass Dasher eine relativ gute Leistung im Vergleich zu den anderen Techniken hat. Die Steigung der Leistung ist relativ steil, dies könnte zu einer gewissen Verbesserung der Geschwindigkeit bei weiterer Übung führen, was z.B. bei Software-Tastatur Eingabe wie Qwerty-Tapping nicht der Fall wäre, da die Steigung gering ist.

3 Erweiterung von tastaturbasierte Eingabe: Presense

3.1 Motivation

Das Modell der tastaturbasierter Texteingabe bezieht sich auf Tasten mit 2 Zuständen. Jede Taste kann entweder im Zustand gedrückt oder nicht gedrückt sein. Dies erlaubt aber einen relativen einfachen Interaktionsmuster.

Ausserdem ist es in gewissen Situationen interessant, eine Vorschau des Effektes eines Tastendruckes zu erhalten. Insbesondere wenn der Effekt des Drückens nicht rückgängig gemacht werden

kann — z.B. in der realen Welt, um das Lernverhalten zu optimieren (die Berührung einer Taste zeigt direkt die Vorschau des Effektes), oder um Platz auf dem Bildschirm zu sparen (nur relevante Informationen werden angezeigt).

3.2 Technische Realisierung

Jede Taste wird durch 3 Zustände modelliert. Der zusätzliche Zustand ist die Berührung einer Taste. Dies ermöglicht zugleich Bewegungen auf der Tastatur zu detektieren.

Jede Tastenoberfläche besteht aus Kupfer. Der kapazitive Effekte des menschlichen Körpers wird dadurch gemessen. Ein Atmel 8-Bit Mikroprozessor ermöglicht 50 mal pro Sekunde 25 Tasten zu scannen.

3.3 Anwendungsbeispiel 1: Inhalt Browsing

PreSense erlaubt eine relativ einfache Navigation durch eine hierarchisch geordnete Informationsmenge. Das Fenster wird in verschiedene Zonen unterteilt, wobei jede Zone genau einer Taste zugeordnet wird. Die Berührung einer Taste zeigt auf dem Bildschirm ein Pop-up mit der Vorschau des Inhalts der nächsten Stufe von der Kategorie die gewählt wird. Das Drücken einer Taste geht in die nächste Stufe über. Das System ist besonders geeignet fürs Browsing einer Informationsmenge wie ein Filesystem oder ein Fotoalbum.

3.4 Anwendungsbeispiel 2: Texteingabe

3.4.1 Previsualisierbare Tastatur

Diese Anwendungsmöglichkeit ist insbesondere relevant für sehr kleine Tastaturen, wo es nicht genug Platz auf der Taste gibt, um die Buchstaben anzubringen. Bei der Berührung einer Taste erscheint auf dem Bildschirm die Tastatur und die berührte Taste wird angezeigt. Das System stellt dabei die Idee des On-demand Interfaces dar. Es ist auch für universelle Fernbedienungen nützlich, wo eine Taste je nach Modus eine andere Bedeutung aufweist, was die Vorschau funktionalität wünschbar macht.

3.4.2 Touch-Shift Tastatur

Das Konzept der Touch-Shift Tastatur ermöglicht 2 Zonen innerhalb einer Tastatur zu definieren. Die Berührung einer Taste einer Zone wechselt den Modus der zweiten Zone (siehe Abbildung 8(a)). Falls nur die erste Zone für Berührungsinteraktionen vorgesehen ist erlaubt es insgesamt $(N + 1) \times M$ Interaktionsmöglichkeiten, wobei N die Anzahl Tasten der ersten Zone und M der zweiten Zone darstellen. In dem Fall wird auch die Vorschau benutzt, da die Bedeutung jeder Taste nicht eindeutig ist.

3.5 Anwendungsbeispiel 3: Detektion von Bewegungsmustern

3.5.1 Kontaktmuster

PreSense erlaubt es, Kontaktmuster zu detektieren. Sei es von einem oder mehreren Fingern oder von der ganzen Handfläche. Ein Muster kann einen Befehl auslösen oder den Modus der Tastatur wechseln.

3.5.2 Fingerbewegung

Fingerbewegungen können ebenfalls detektiert werden. Es kann z.B. erkannt werden, ob 2 Finger quer durch die Tastatur in eine Richtung verschoben werden. Eine andere Interaktionsmöglichkeit besteht darin, zyklische Bewegungen für Inhaltsverschiebung zu verwenden, ähnlich wie das kreisförmige Scrolling vom iPod (siehe Abbildung 8(b)).

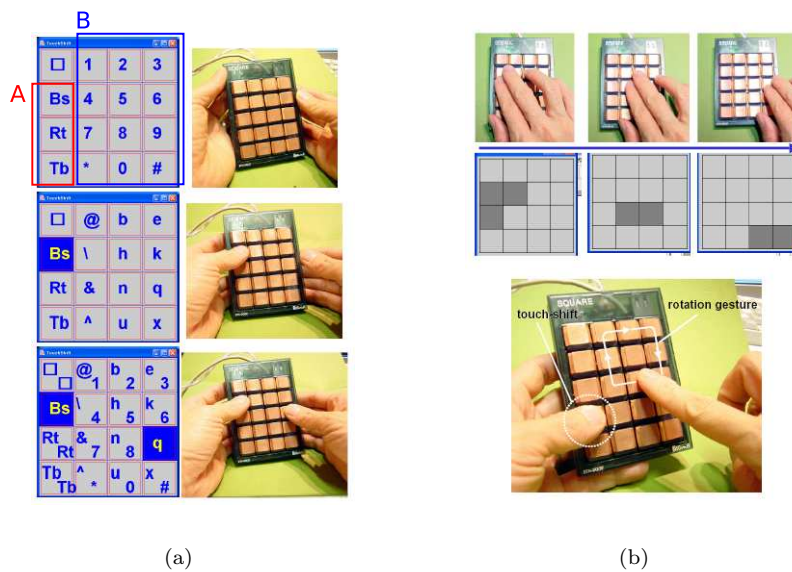


Abbildung 8: (a) Touch-Shift Interaktion. Die Berührung einer Taste der Zone A ermöglicht den Modus der Zone B zu wechseln (b) Erkennung von Bewegungsmustern. Oben: die Verschiebung zweier Finger, unten: zyklische Fingerbewegungen können als Inhaltsverschiebung verwendet werden.

4 Zusammenfassung

In dieser Arbeit haben wir verschiedene Systeme kennengelernt, die unterschiedliche Leistungen und Anwendungsmöglichkeiten besitzen. Eine Übersicht der Leistungen der behandelten Verfahren befindet sich in der Tabelle 1.

Tabelle 1: Übersicht der Leistungen der behandelten Verfahren.

Tippmethode	Leistung [wpm]
Mündliche Sprache	~200
Handschrift	~20
PC-Tastatur	50-70
Multitap	20-25
T9 ³	40-45
TiltText	~13
Halb-Qwerty	40-50
Graffiti	20
Unistrokes ⁴	~30
Lightglove	-
Thumbwheel	-
Dasher	~20
PreSense	-

³ideale Bedingungen

⁴nicht systematisch getestet

5 Referenzen

Literatur

- [MacKen02] MacKenzie and Soukoreff, *Text Entry for Mobile Computing: Models and Methods, Theory and Practice*, 2002
- [Dasher00] David J. Ward, Alan F. Blackwell and David J. C. MacKay, *Dasher - a Data Entry Interface Using Continuous Gestures and Language Models*. UIST '00, 2000.
- [Rekim03] J. Rekimoto et al. *PreSense: Interaction Techniques for Finger Sensing Input Devices*, 2003
- [How01] B. Howard and S. Howard. *Lightglove: Wrist-Worn Virtual Typing and Pointing*, 2001
- [Wigd03] D. Wigdor and R. Balakrishnan, *TiltText: Using Tilt for Text Input to Mobile Phones*, 2003
- [May98] G. J. Mayer-Kress, K. M. Newell, Y.-T. Liu. *What Can We Learn From Learning Curves?* 1998, <http://www.personal.psu.edu/faculty/g/x/gxm21/NECSI98/>
- [MacKen01] MacKenzie et al. *LetterWise: Prefix-based Disambiguation for Mobile Text Input*, 2001
- [Fle] M. D. Fleetwood et al. *An Evaluation of Text-Entry in Palm OS Graffiti and the Virtual Keyboard*
- [Tara03] Peter Tarasewich. *Evaluation of Thumbwheel Text Entry Methods*, 2003
- [Shan93] C. E. Shannon. *Collected Papers*. IEEE Press, New York, 1993. Edited by N. J. A. Sloane and A. D. Wyner.
- [Dru92] C. G. Drury and E. R. Hoffmann. A model for movement time on data-entry keyboards. *Ergonomics*, 35(2):129-147, 1992.