Fachseminar - Verteilte Systeme

24. Mai 2005

Patrick Jayet

Alternative Texteingabe für Mobilgeräte

Motivation

- Warum ist alternative Texteingabe wichtig?
  - Zunehmende Anzahl von mobilen Geräten (Handies, PocketPC, PDA, ...)
  - Killer Applikationen, die Texteingabe brauchen (SMS, E-Mail, Agenda, usw.)
  - Oft begrenzter Platz
- PC Tastatur kann nicht integriert werden!

Ziele des Vortrages

- Einen Überblick über den Bereich gewinnen
  - Problematik bei der Bewertung eines Verfahrens
  - Existierende Texteingabe
- Einblick in 2 aktuelle Entwicklungen

Übersicht

- Einführung
  - Bewertung eines Eingabeverfahrens 
  - Tastaturbasierte Eingabe
    - Tastatur eines Mobiltelefons (Multitap, T9, Tilttext)
    - Halb-Qwerty
  - Stylus-basierte Eingabe (Graffiti, Unistrokes)
  - Unkonventionelle Eingabe (Lightglove, Thumbwheel)
- Hybride Texteingabe: Dasher
- Erweiterung von Tastaturbasierter Eingabe: PreSense

Texterzeugung vs Textkopie I

- die Geschwindigkeit eines Systems bewerten – idealerweise mit Texterzeugung
- Aber keine Kontrolle über:
  - die Verteilung der Buchstaben
  - Ausserdem: Text ausdenken braucht Zeit, es kann das Resultat beeinflussen

➡ im allgemeinen wird Textkopie verwendet

Texterzeugung vs Textkopie II

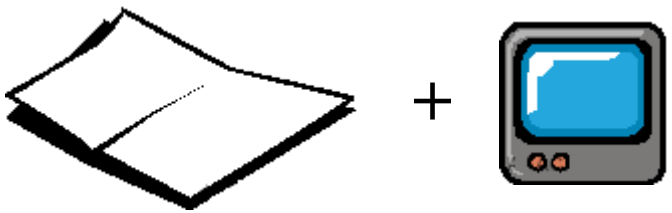
- Nachteil von Textkopie:
  - Der Benutzer muss der geschriebene Text auch lesen

  ➡ 1 Focus of Attention (FOA) mehr

- Bsp. PC Tastatur

fortgeschrittener Benützer:
1 FOA

Anfänger:
2 FOA

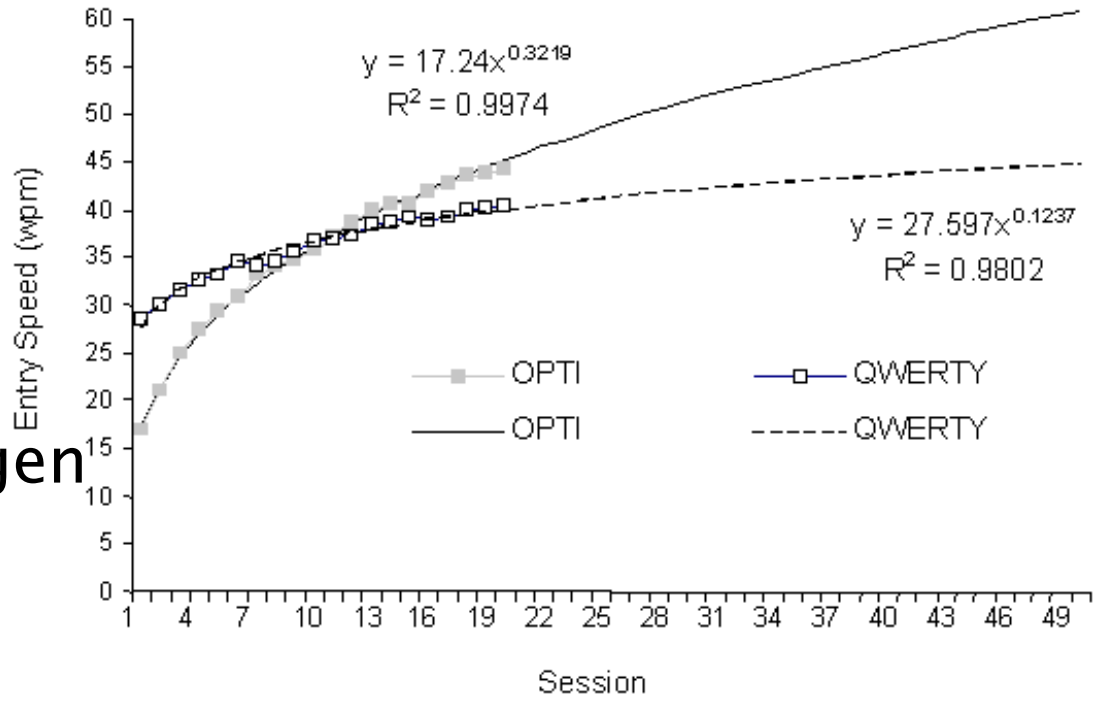


Anfänger vs Fortgeschrittene

- Ziel eines guten Eingabesystems
  - Anfänger erreichen rasch eine akzeptabele Leistung
  - Fortgeschrittene haben eine gute Leistung
- Modellierung durch das Potenz-Gesetz des Lernens

$$P(n) = A + B \cdot n^{\alpha}$$

P(n): Leistung
n: # Wiederholungen
A, B, α: Konst.



Einheit der Geschwindigkeit

- Zeichen pro Sekunde (cps)
- Wörter pro Minute (wpm)
  - wobei angenommen wird, dass 1 Wort aus 5 Zeichen besteht (unabhängig davon, ob es Zeichen oder Leerzeichen sind)

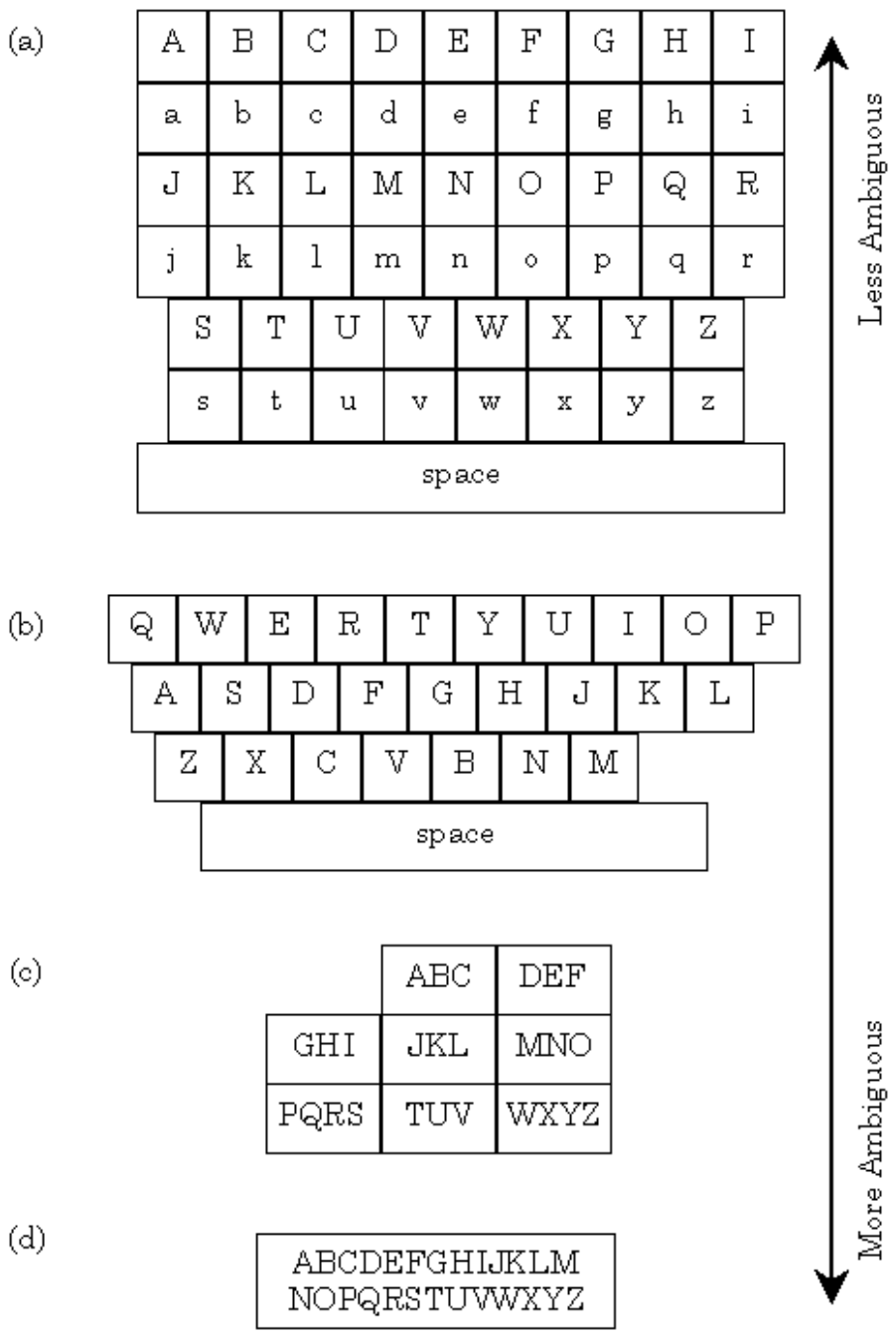
$$G_{wpm} = G_{cps} \cdot 60/5 = G_{cps} \cdot 12$$

Übersicht

- Einführung
  - Bewertung eines Eingabeverfahrens
  - Tastaturbasierte Eingabe
    - Tastatur eines Mobiltelefons (Multitap, T9, Tilttext)
    - Halb-Qwerty
  - Stylus-basierte Eingabe (Graffiti, Unistrokes)
  - Unkonventionelle Eingabe (Lightglove, Thumbwheel)
- Hybride Texteingabe: Dasher
- Erweiterung von Tastaturbasierter Eingabe: PreSense

Mehrdeutigkeit der Tasten

- Wird als Kontinuum wahrgenommen
- 2 interessante Fälle sind (b) und (c)

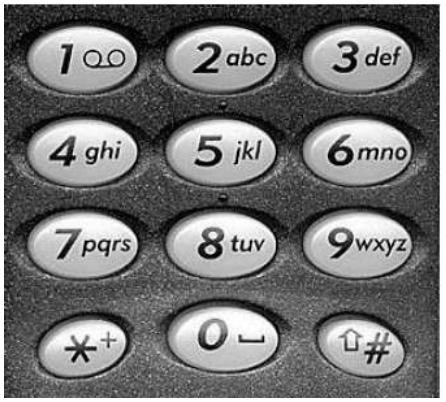


Anhaltspunkt 1: Tastatur eines Mobiltelefons

- Tasten 2–9 für Buchstaben verwendet, 3–4 Zeichen pro Taste
- Mehrdeutigkeit der Tasten muss aufgelöst werden

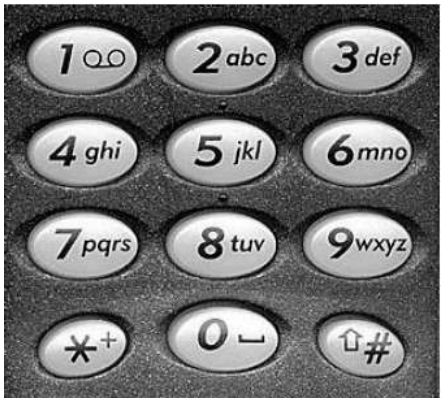
➡ Multitap, T9, Tilttext...

Multitap



- Meistbenutzte Methode
- Mögliches Problem:
  - Aufteilung einer Sequenz (ON, 3+2-mal Taste 6)
  - ➡ Trennung durch Timeout oder Timeout-kill Taste
- Leistung: 20-25 wpm (Fortgeschrittene) (Vergleich Comp.-Tastatur: 50-70 wpm)

T9



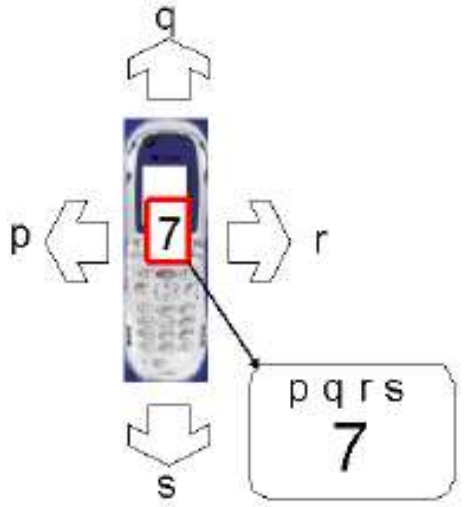
- Wörterbuch wird zur Auflösung von Mehrdeutigkeit der Tasten verwendet
- Leistung (Fortgeschrittene) von 40–45 wpm*

  (Vergleich Comp.-Tastatur: 50–70 wpm)
- Leistung nimmt aber schnell ab, falls spezielle Wörter verwendet werden, oft der Fall bei realen Anwendungen (Slang, Eigenname, usw.)

* Annahme: alle Wörter im Worterbuch, keine Mehrdeutigkeit

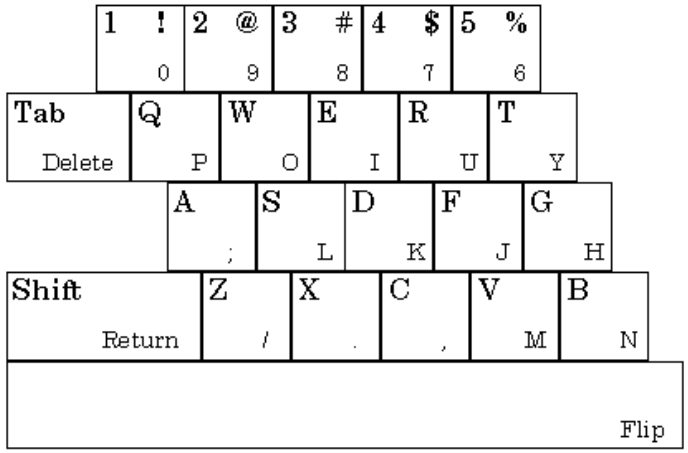
TiltText

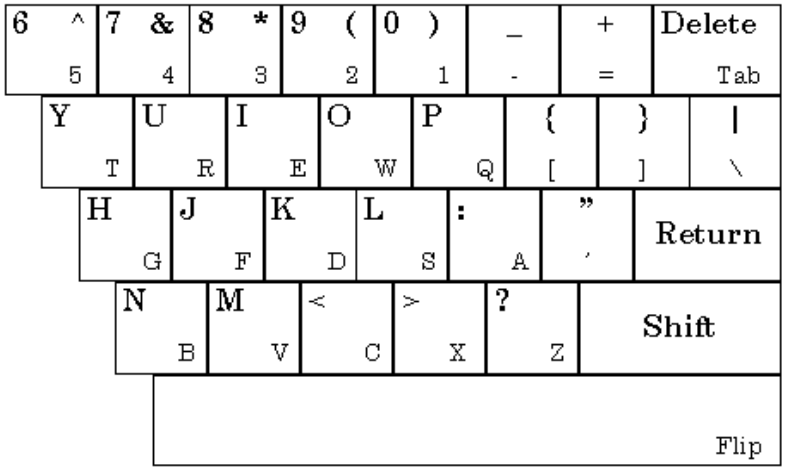
- Auflösung der Mehrdeutigkeit durch Kippen des Mobiltelefons
- Leistung etwa 20% schneller als für Multitap
- Nachteil: Fehlerrate wesentlich höher (2 bis 3 mal höher)



Anhaltspunkt 2: Halb-Qwerty Tastatur

- 2 Hälften der Tastatur durch Drücken der Leertaste gewählt
- Entweder von linker oder rechter Hand gebraucht
- Leistung zw. 41–73% der vollständigen Tastatur nach 10 Stunden Praxis





Übersicht

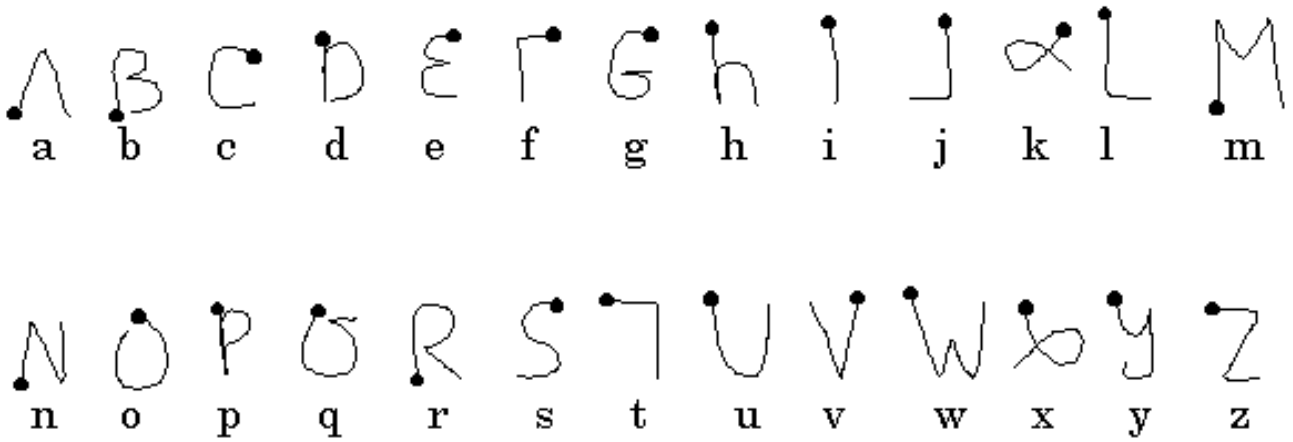
- Einführung
  - Bewertung eines Eingabeverfahrens
  - Tastaturbasierte Eingabe
    - Tastatur eines Mobiltelefons (Multitap, T9, Tilttext)
    - Halb-Qwerty
  - Stylus-basierte Eingabe (Graffiti, Unistrokes)
  - Unkonventionelle Eingabe (Lightglove, Thumbwheel)
- Hybride Texteingabe: Dasher
- Erweiterung von Tastaturbasierter Eingabe: PreSense

Stylus-basierte Eingabe

- Erkennung von Stylus-Bewegungen
- Benötigt beide Händen (Nachteil)
- 2 Hauptprobleme:
  - Segmentierung (Erkennung der Grenze zw. 2 Zeichen)
  - Erkennung der Buchstaben

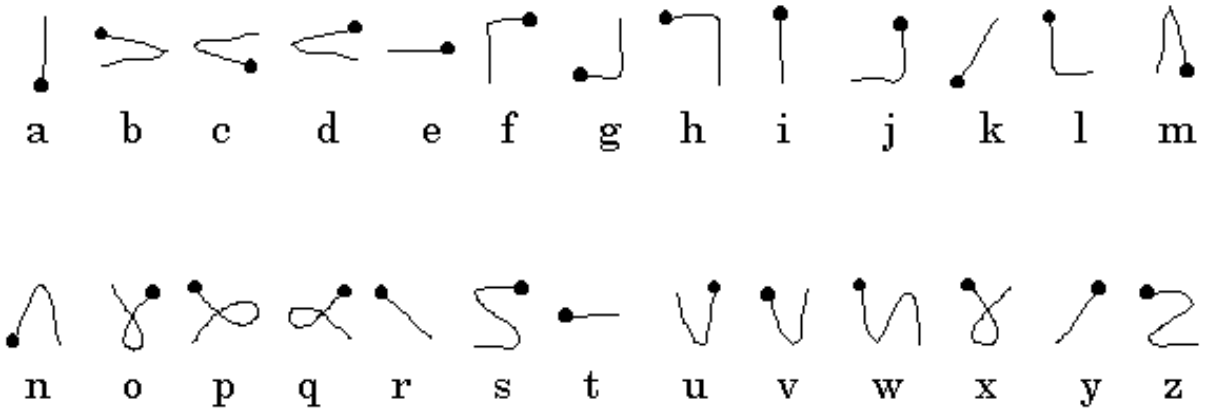
➡ Verschiedene Ansätze, u.a. Graffiti, Unistrokes

Graffiti



- Vereinfachte Version des Alphabets
  - behält aber die Ähnlichkeiten mit Handschrift
  - ➡ relativ schnell lernbar
- Jeder Buchstabe aus einem einzigen Strich
- Signifikanter Faktor beim Erfolg des Palm PDA
- Leistung von 20 wpm (Fortgeschrittene) und 7 wpm (Anfänger)

Unistrokes



- Xerox PARC 1993
- Rein synthetischer Ansatz: keine Ähnlichkeit mit Handschrift
- Leistung etwa 34 wpm (Tests nicht systematisch, ohne die Fehler bei der Eingabe zu berücksichtigen)

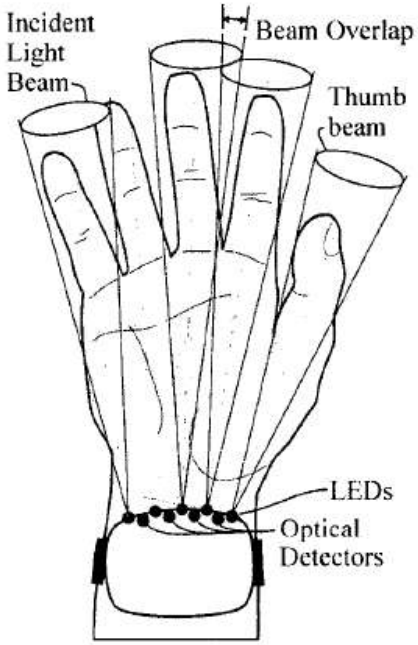
➡ Kein Erfolg

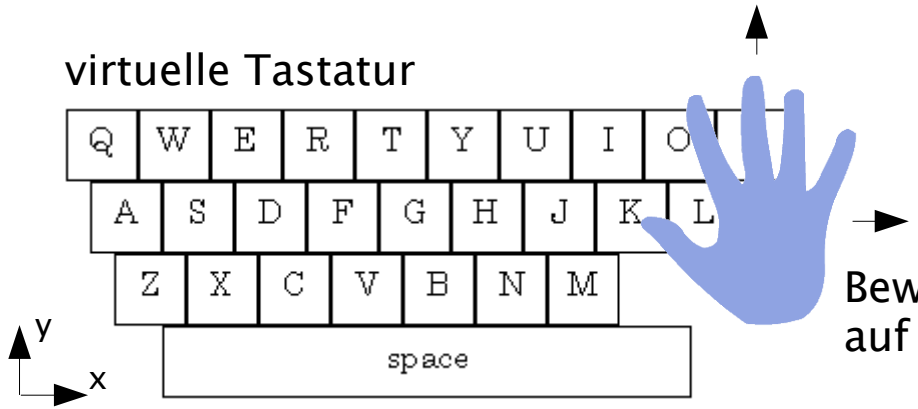
Übersicht

- Einführung
  - Bewertung eines Eingabeverfahrens
  - Tastaturbasierte Eingabe
    - Tastatur eines Mobiltelefons (Multitap, T9, Tilttext)
    - Halb-Qwerty
  - Stylus-basierte Eingabe (Graffiti, Unistrokes)
  - Unkonventionelle Eingabe (Lightglove, Thumbwheel)
- Hybride Texteingabe: Dasher
- Erweiterung von Tastaturbasierter Eingabe: PreSense

Unkonventionelle Eingabe

- Lightglove
  - Abstandssensoren für die Detektion der Fingerbewegung (IR Sensoren)
  - Bewegungssensoren für die Berechnung der absoluten Position des Lightglove im Raum
  - Handlung auf einer virtuellen Tastatur





Bewegung der Hand auf der virtuellen Tastatur

Unkonventionelle Eingabe

- Thumbwheel
  - Ein Drehrad (X-Position)
  - Eine Taste (auf dem Drehrad drücken)
  - Wahl einer Buchstabens in einer Liste (hierarchisch oder linear)
  - 8-10 Tastedruck pro Zeichen

(space)

A	→	ABCDEFGHIJKLM
N	→	NOPQRSTUVWXYZ
0	→	0123456789
.	→	.?!

Hierarchical Method Menu Structure

_ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789.?!

Loop Method Character Set

Übersicht

- Einführung
  - Bewertung eines Eingabeverfahrens
  - Tastaturbasierte Eingabe
    - Tastatur eines Mobiltelefons (Multitap, T9, Tilttext)
    - Halb-Qwerty
  - Stylus-basierte Eingabe (Graffiti, Unistrokes)
  - Unkonventionelle Eingabe (Lightglove, Thumbwheel)
- Hybride Texteingabe: Dasher
- Erweiterung von Tastaturbasierter Eingabe: PreSense

Motivation

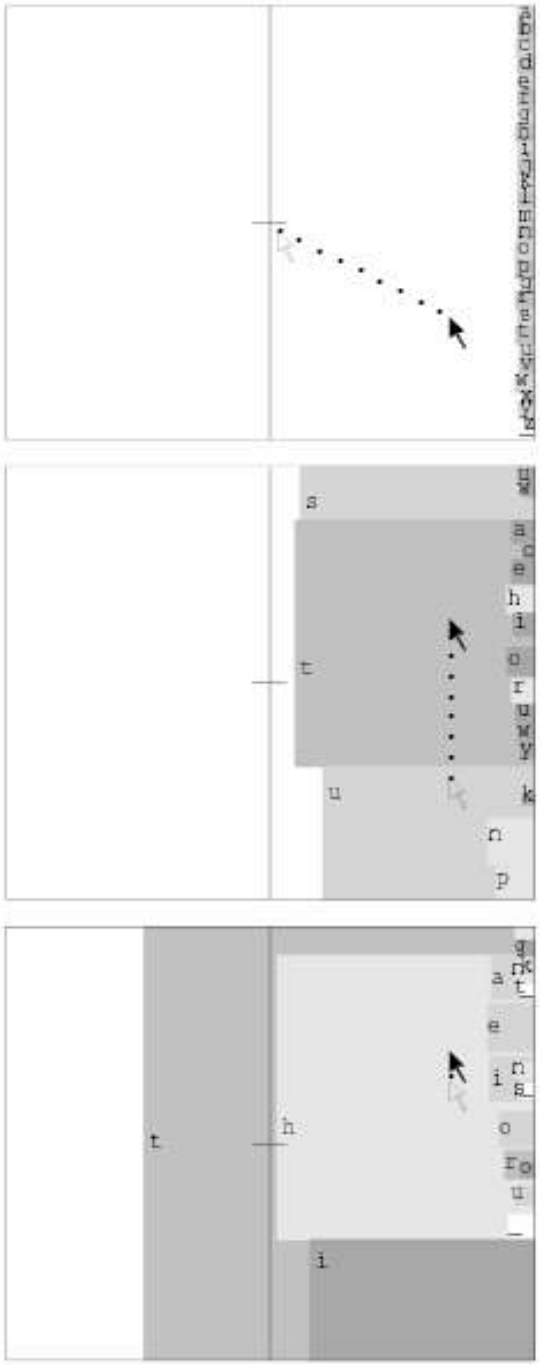
- Für eine Qwerty-Tastatur: eine Bewegung ~6.3 Bits (eine von 80 Tasten gewählt)
- Entropie pro Buchstabe in englischem Text wurde von Shannon auf 1 Bit geschätzt
- Dasher basiert auf einer kontinuierlichen Bewegung (Grenze etwa 14 Bits pro Sek., so Drury, Hoffmann)

  ➡ theoretisch 14 Zeichen pro Sek. (stimmt aber nicht)
- Begrenzung der Rate von Bewegungereignissen für einen Menschen

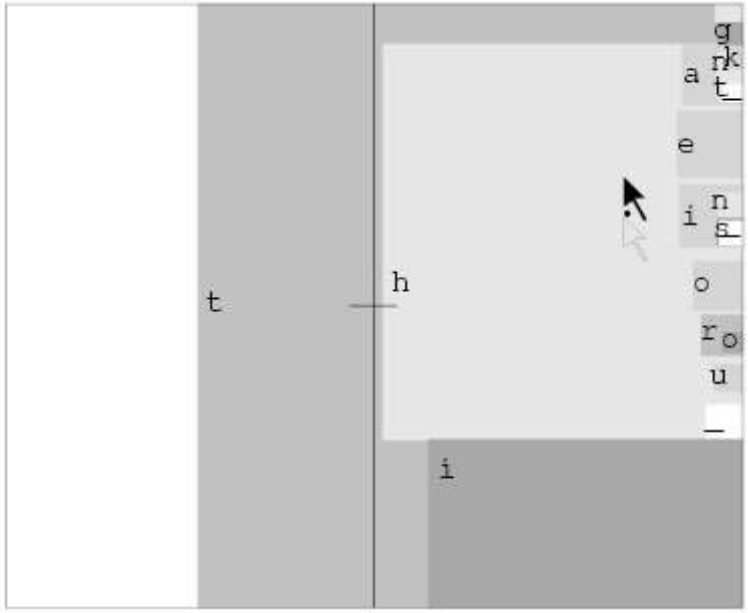
  ➡ obere Schanke für die Tippgeschwindigkeit

Einführung

- Unkonventionnelle Texteingabe
- Basiert auf Sprachmodellierung (W'keit von Zeichenfolgen innerhalb einer Sprache, nicht Wörterbuch-basiert)
- 1997 von MacKay et al. entwickelt (kontinuierliche Entwicklung bis jetzt)
- Ziel: Eingabe für mobile Geräte und bewegungsbehinderte Personen (mittels Eyetracking)

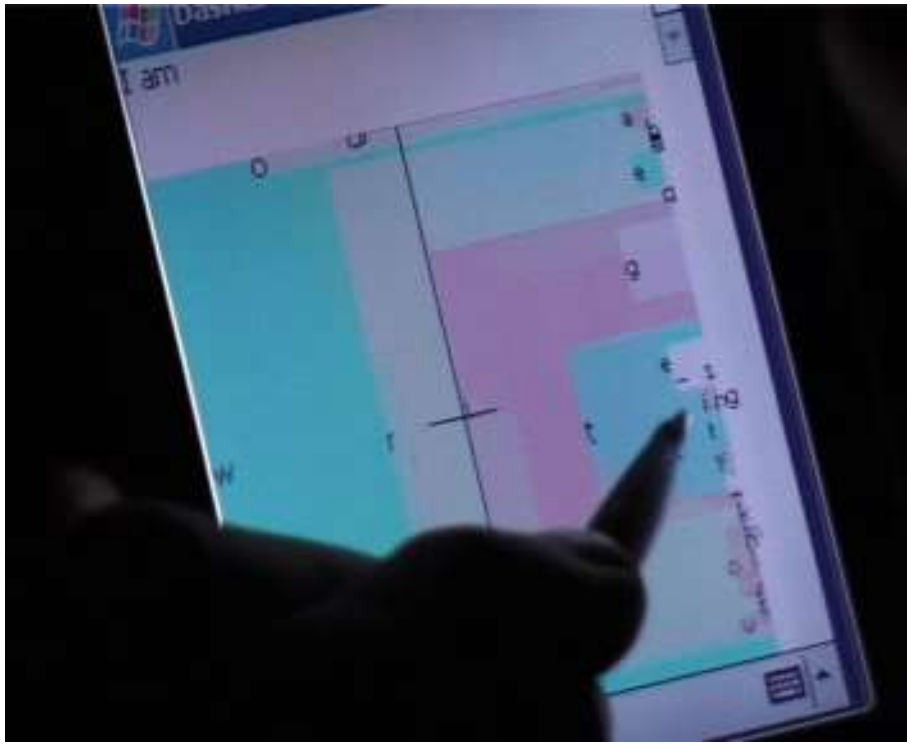


Prinzip

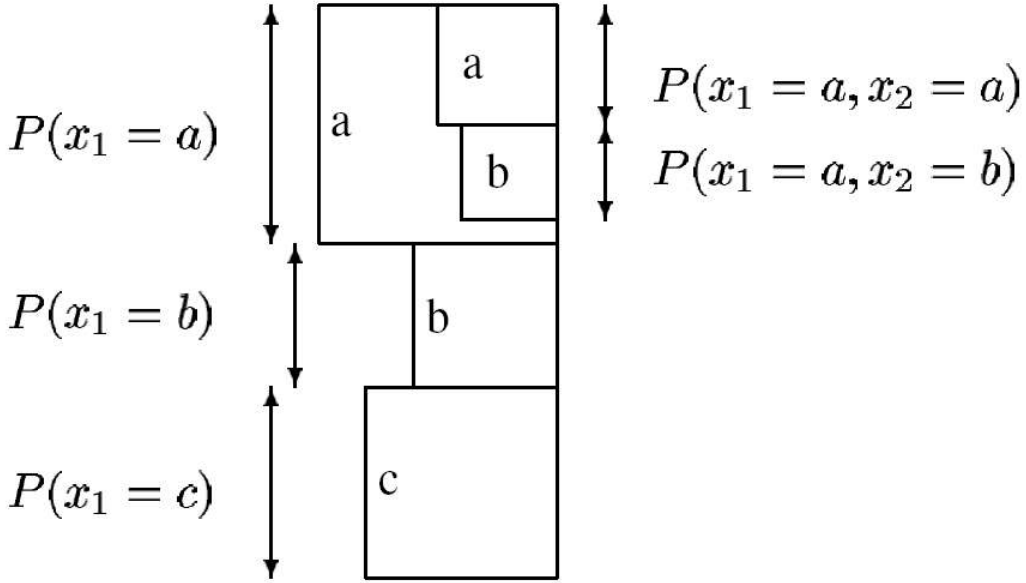


- Höhe eines Buchstabens entspricht seiner W'keit in der Sprache gegeben die vorherige Zeichenkette
- durch 2-dimensionale Bewegung geführt:
  - X-Bewegung: Geschwindigkeit des Scrolling
  - Y-Bewegung: Wahl eines Buchstabens
- Zeichen wird gewählt, wenn es die Mittelinie des Bildschirms überquert

Dasher: Kurzfilm



Höhe eines Buchstabens



- Berechnung der Höhe (Bayes):

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

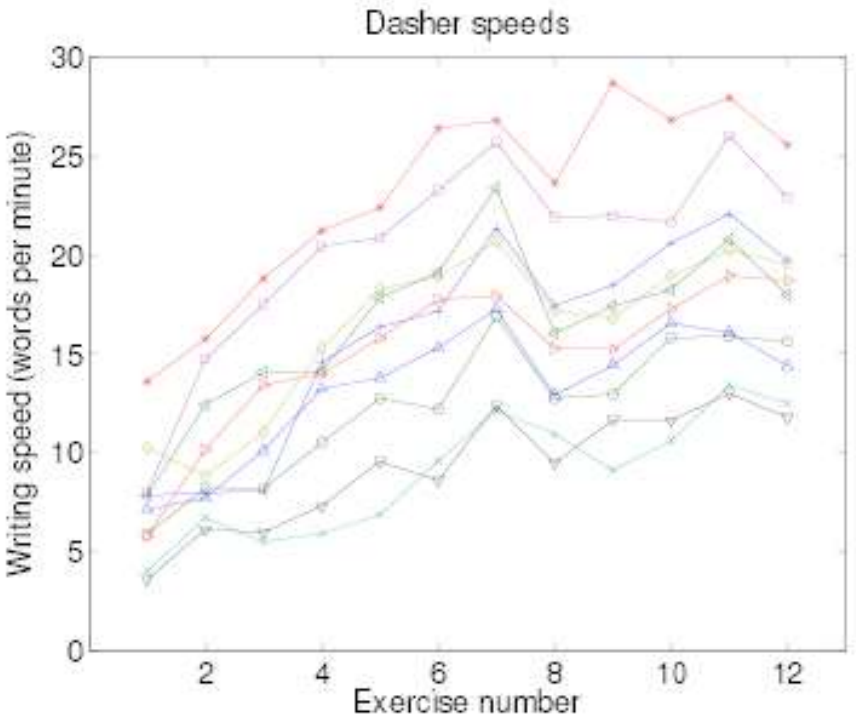
- Minimale W'keit für jeden Buchstaben von δ (z.B. $\delta=0.02$)

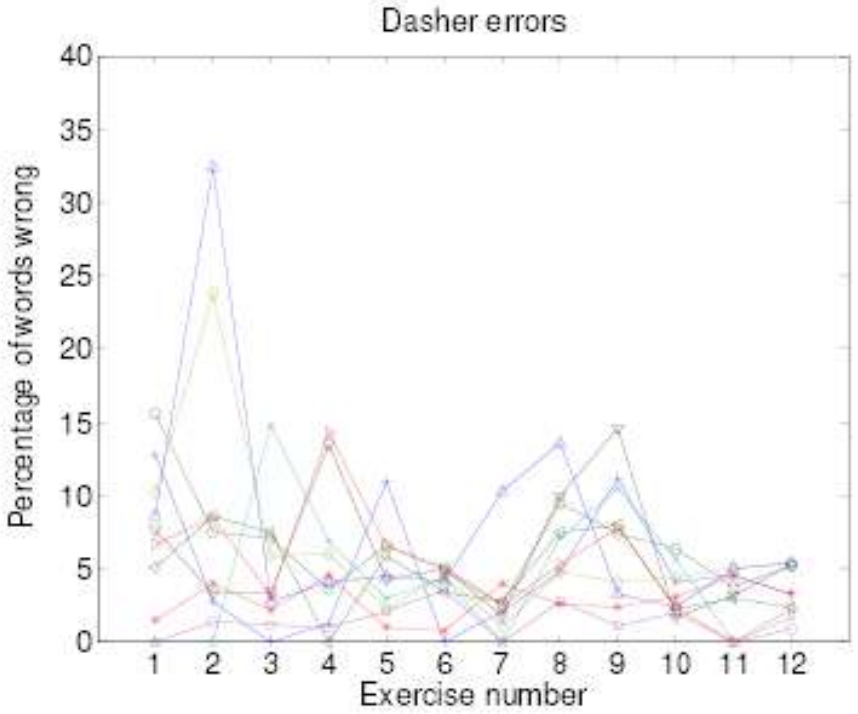
  ➡ damit seltene Buchstaben gefunden werden können

Sprachmodellierung

- Das benutzte Textmodellierungsverfahren heisst Prediction by Partial Match (PPM)
  - Abschätzung der W'keit eines Zeichens, gegeben ein Kontext von ein paar Zeichen
- Es erlaubt englische Texte auf ~2 Bits pro Zeichen zu komprimieren/codieren
- Gewählt, da relativ effizient und wirksam
  - Die W'keit des nächsten Buchstabens muss bei jeder Eingabe neu berechnet werden

Ergebnisse

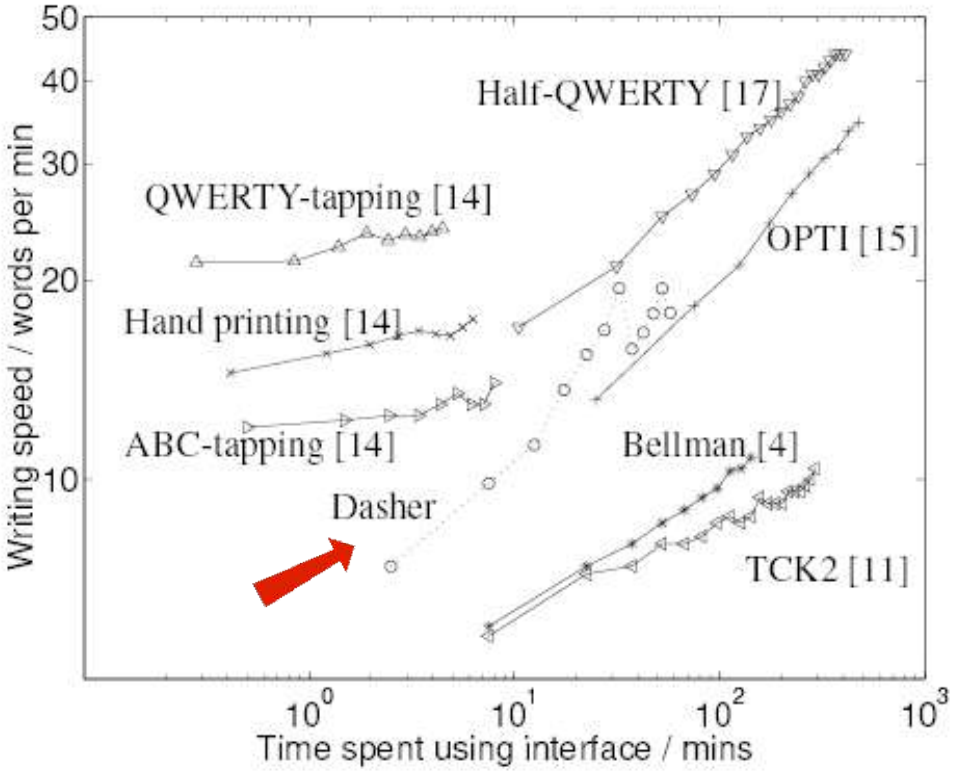




- Modelliert durch das Potenz-Gesetz des Lernens
- Fortgeschrittener Benutzer erreicht etwa 20 wpm (etwa wie Multitap, bedienbar aber mittels Eyetracking)
- Fehlerrate relativ gering (unter 5% der Wörter)

Vergleich mit anderen Techniken

- Relativ gute Geschwindigkeit im Vergleich zu den anderen Techniken
- Relativ steile Steigung
  - evtl. Verbesserungs-potential



Übersicht

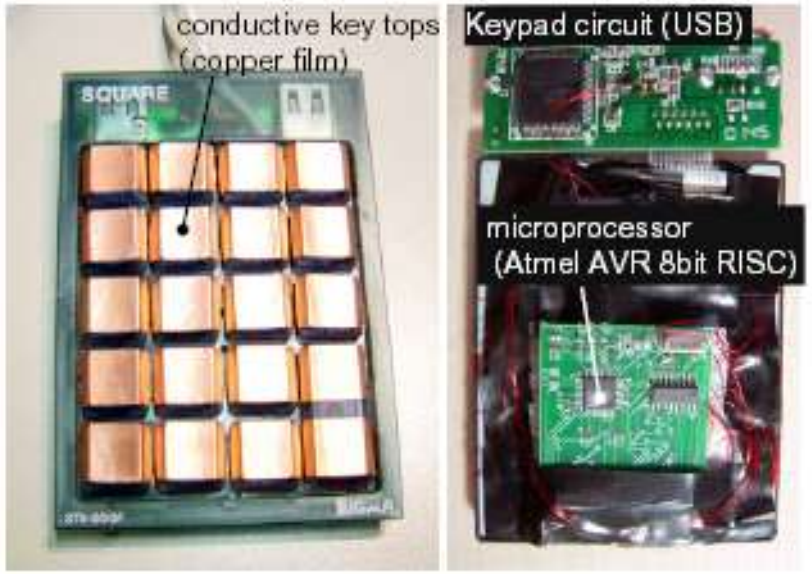
- Einführung
  - Bewertung eines Eingabeverfahrens
  - Tastaturbasierte Eingabe
    - Tastatur eines Mobiltelefons (Multitap, T9, Tilttext)
    - Halb-Qwerty
  - Stylus-basierte Eingabe (Graffiti, Unistrokes)
  - Unkonventionelle Eingabe (Lightglove, Thumbwheel)
- Hybride Texteingabe: Dasher
- Erweiterung von Tastaturbasierter Eingabe: PreSense

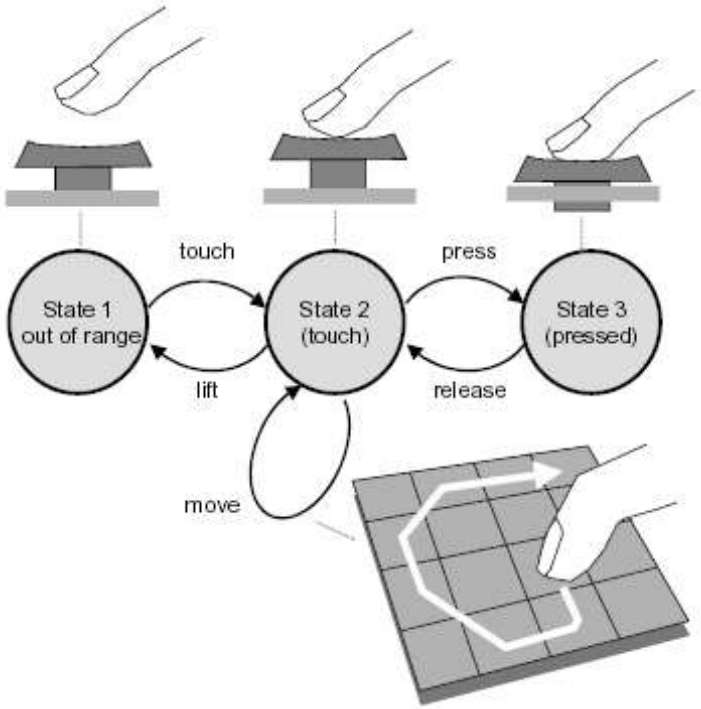
Motivation

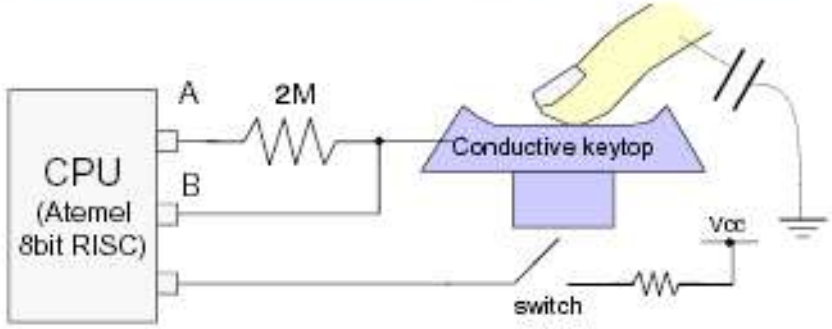
- Übliche Tasten:
  nur 2 Zustände
  ➡ Interaktionsmuster relativ einfach
- In gewissen Fällen ist eine Vorschau einer Operation erwünscht:
  - Effekt des Drückens einer Taste kann nicht rückgängig gemacht werden (z.B. in der realen Welt)
  - Effizientere Lernstrategien (die Berührung einer Taste zeigt eine Vorschau der Wirkung, wie beim Tooltip)
  - Platz auf dem Bildschirm sparen (nur relevante Informationen zeigen)

Technische Realisierung

- 3 Zustände
- Ermöglicht Bewegungsdetektion







Anwendungsbeispiel 1

Inhalt-Browsing

- Hierarchisches Browsing
  - Fenster in Zonen unterteilt (1 für jede Taste)
  - Berührung einer Taste ➡ Pop-up mit Vorschau der nächsten Stufe
  - Drücken: in die nächste Stufe übergehen
  - Für Browsing von Verzeichnissen und Dateien geeignet

Anwendungsbeispiel 2

Texteingabe

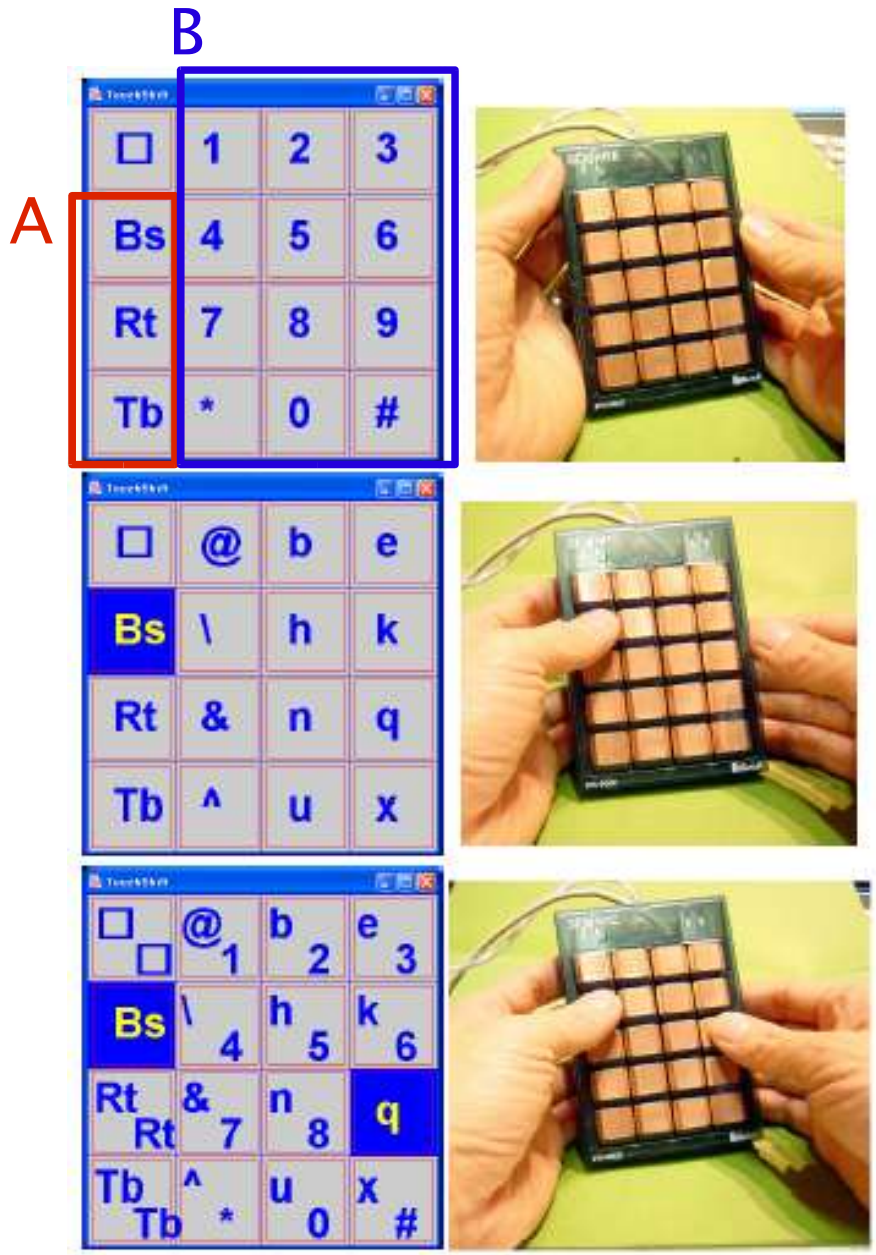
- Tastatur mit Vorschau
  - Berührung einer Taste
    ➡ Tastatur angezeigt und Taste hervorgehoben
  - Nützlich für kleine Tasten (nicht genug Platz um Zeichen anzubringen)
  - On-Demand Interface: Tastatur auf dem Bildschirm nur gezeigt, wenn nötig

- Für Multifunktions-tastatur: Vorschau der Wirkung

Anwendungsbeispiel 2

Texteingabe

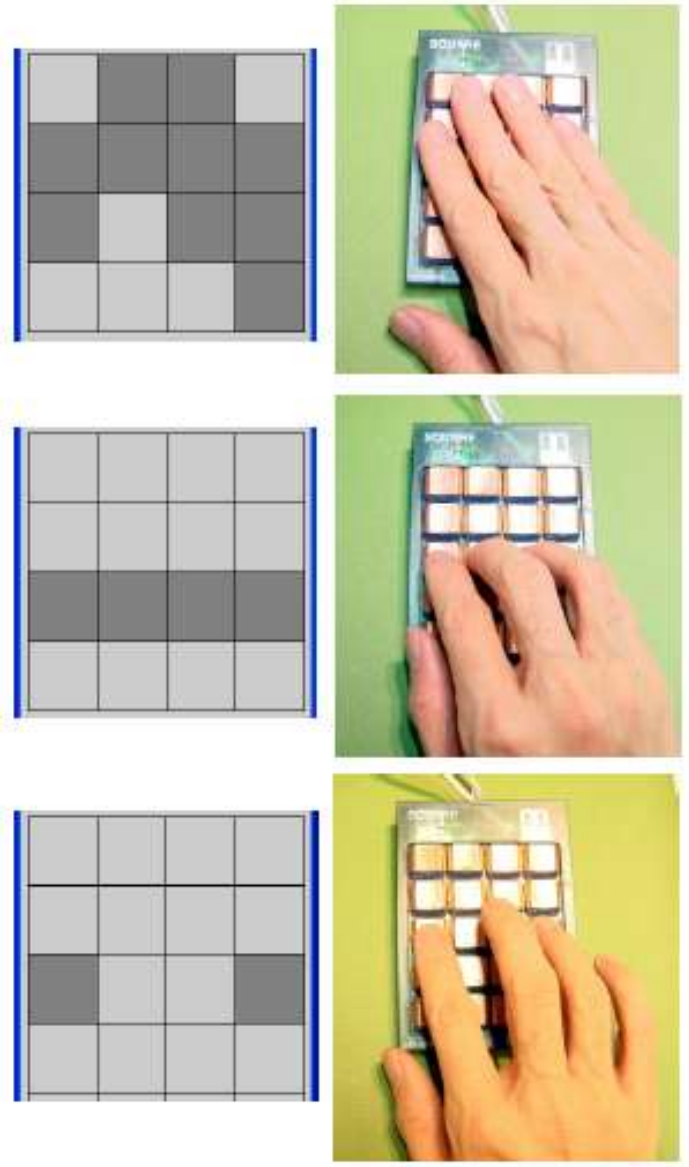
- Touch-Shift Tastatur
  - Berührung einer Taste der Zone A ändert den Modus von Teil B
  - (N+1) x M Interaktions-möglichkeiten
    N: # Tasten Teil A
    M: # Tasten Teil B
  - Hier auch: Vorschau der Selektion



Anwendungsbeispiel 3

Detektion von Bewegungsmustern

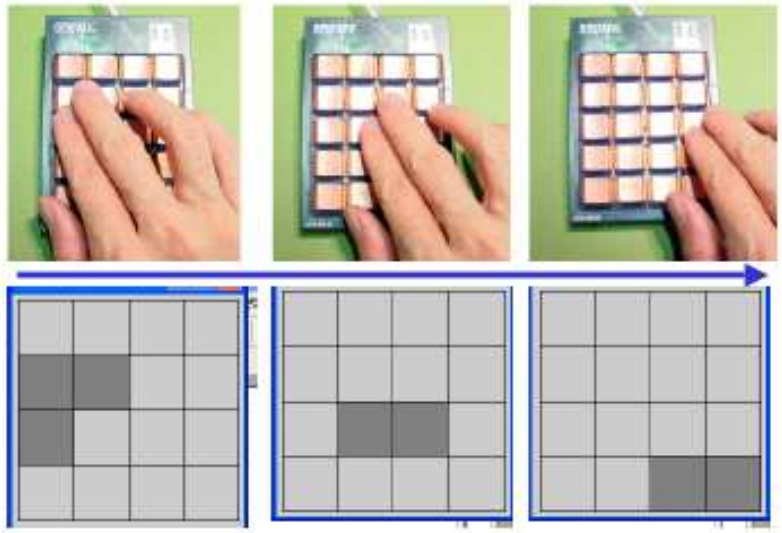
- Kontaktmuster
  - Handfläche
  - Finger
- Ein Muster löst einen Befehl aus oder wechselt den Modus

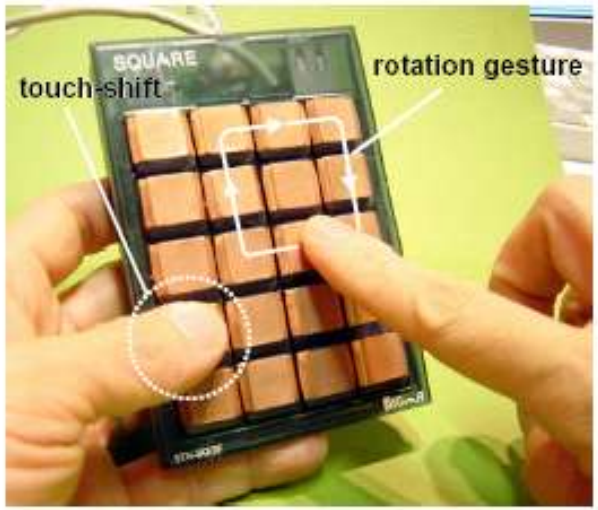


Anwendungsbeispiel 3

Detektion von Bewegungsmustern

- Fingerbewegung
  - Bewegung z.B. von 2 Fingern
  - Touch-Shift: Berührung einer Taste aktiviert einen Modus
  - Inhaltsverschiebung durch zyklische Bewegung (ähnlich wie iPod scroll)





Fazit

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

* ideale Bedingungen
** 20% schneller als seltsame Multitap-Impl.
*** nicht systematisch getestet

Referenzen I

- MacKenzie and Soukoreff, *Text Entry for Mobile Computing: Models and Methods, Theory and Practice*, 2002
- 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.
- J. Rekimoto et al. *PreSense: Interaction Techniques for Finger Sensing Input Devices*, 2003
- B. Howard and S. Howard. *Lightglove: Wrist-Worn Virtual Typing and Pointing,* 2001

Referenzen II

- D. Wigdor and R. Balakrishnan, *TiltText: Using Tilt for Text Input to Mobile Phones, 2003*
- 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/
- MacKenzie et al. *LetterWise: Prefix-based Disambiguation for Mobile Text Input*, 2001
- M. D. Fleetwood et al. *An Evaluation of Text-Entry in Palm OS Graffiti and the Virtual Keyboard*
- *Peter Tarasewich. Evaluation of Thumbwheel Text Entry Methods*, 2003

Das war's.

Fragen, Bemerkungen?