

Universität Leipzig  
Institut für Informatik  
Lehrstuhl für Angewandte Telematik / e-Business  
Prof. Dr. Volker Gruhn

UNIVERSITÄT LEIPZIG



Seminararbeit

## **Allgemeine Klassifikation von Texteingabe-Techniken in mobilen Endgeräten**

**User Interface Engineering für mobile und web-basierte Anwendungen  
(SS 2005)**

Autor: Christian Tilgner (Mat.-Nr. 9639106)  
Studiengang: Informatik - Master (2. Semester)

Betreuer: Tobias Brückmann

Eingereicht am: 24.06.2005



## **Inhaltsverzeichnis**

<b>Inhaltsverzeichnis .....</b>	<b>3</b>
<b>Abkürzungsverzeichnis .....</b>	<b>4</b>
<b>Abbildungsverzeichnis .....</b>	<b>5</b>
<b>1 Einleitung.....</b>	<b>6</b>
<b>2 Problemstellung .....</b>	<b>7</b>
<b>3 Klassifikation von Eingabeverfahren .....</b>	<b>8</b>
3.1 Stylus-basierte Eingabemöglichkeiten .....	8
3.1.1 Handschrifterkennung .....	9
3.1.1.1 Worterkennung .....	9
3.1.1.2 Zeichenerkennung .....	9
3.1.2 Soft-Keyboard .....	10
3.1.3 T-Cube .....	11
3.2 12-Button-basierte Eingabemöglichkeiten.....	16
3.2.1 Multitap .....	17
3.2.2 Less-Tap.....	18
3.2.3 Two-Key.....	19
3.2.4 Wörterbuch-basierte Verfahren.....	19
3.2.5 Präfix-basierte Verfahren .....	20
3.2.6 TiltText .....	21
3.2.7 Bewertung.....	24
3.3 Weitere Eingabemöglichkeiten.....	25
3.3.1 Ansteckbare Tastaturen.....	25
3.3.2 Optische Keyboard-Simulation.....	26
<b>4 Zusammenfassung .....</b>	<b>27</b>
<b>Literaturverzeichnis .....</b>	<b>28</b>

## Abkürzungsverzeichnis

KSPC	Keystrokes per character
KSPS	Keystrokes per second
PDA	Personal Digital Assistent
QWERTZ	Name für das Layout deutscher Schreibmaschinen- und Computertastaturen. Bezieht sich auf die ersten 6 Buchstaben der deutschen Tastatur.
QWERTY	Name für das Layout englische Schreibmaschinen- und Computertastaturen. Bezieht sich auf die ersten 6 Buchstaben der englischen Tastatur.
SMS	Short Message Service
WLAN	Wireless Local Area Network
WPM	Words per minute

## Abbildungsverzeichnis

Abbildung 1: Alphabet von Unistroke (a) und Graffiti (b) [KRS00] .....	10
Abbildung 2: Soft Keyboard des XDAII.....	10
Abbildung 3: T-Cube Target [VN94] .....	11
Abbildung 4: Target mit Pie-Menü [VN94] .....	12
Abbildung 5: Feedback bei T-Cube [VN94] .....	13
Abbildung 6: Ergebnisse des Geschwindigkeitsexperiments [VN94].....	14
Abbildung 7: Clusterung der Buchstaben .....	15
Abbildung 8: Inhalte der Target-Zellen [VN94] .....	15
Abbildung 9: Standard 12-Key Keypad nach ITU E161 [WB04].....	17
Abbildung 10: Tastaturbelegung von Less-Tap [PS1] .....	18
Abbildung 11: Tastenbetätigungen pro Buchstabe bei Lesstap [PS1].....	19
Abbildung 12: Betätigung der Next-Taste bei LetterWise [MKS1] .....	21
Abbildung 13: TiltText [WB03].....	22
Abbildung 14: TiltType [PCS1] .....	23
Abbildung 15: Groß- und Kleinschreibung bei TiltText [WB03].....	23
Abbildung 16: Vergleich Eingabeverfahren [MKS1].....	25
Abbildung 17: Virtuelle Tastatur [SIE].....	26

## 1 Einleitung

Im Rahmen der vorliegenden Seminararbeit sollen die verschiedenen Texteingabe-Techniken in mobilen Endgeräten vorgestellt werden. Dazu werden zunächst Arten von mobilen Endgeräten vorgestellt, um die verschiedenen Eingabe-Methoden besser einordnen zu können. Ziel dieser Seminararbeit ist nicht nur das Vorstellen der verschiedenen Interaktionstechniken. Vielmehr sollen diese klassifiziert, bewertet und gegenübergestellt werden. Bei den verschiedenen Verfahren liegt das Hauptaugenmerk auf T-Cube, den 12-Button-Eingabemethoden und dem TiltText-Verfahren.

Durch die rasche Entwicklung und die fortschreitende Technik erscheinen immer mehr mobile Endgeräte auf dem Markt. Das Spektrum von mobilen Endgeräten ist weit gefächert. Es reicht von Notebooks, die vorwiegend als Desktop-Computer-Ersatz dienen, über Tablet-PCs, Organizern, PDAs, Smartphones bis zum Mobiltelefon. Diese vielen verschiedenen Mobil-Geräte unterscheiden sich in ihren Eigenschaften sehr stark. Als offensichtlichstes Merkmal ist die Gerätegröße zu nennen. Wo bei Notebooks eine Tasche zum Transport benötigt wird, können Organizer und PDAs schon in der Jacken- bzw. Jackettasche transportiert werden. Smartphones und Mobiltelefone können hingegen bequem in der Hosentasche getragen werden. Als zweites Merkmal ist der Funktionsumfang zu nennen. Wo ein Notebook fast alle Funktionen und Möglichkeiten besitzt, die ein Desktop-Computer auch aufweist, beschränkt sich die Funktionen eines normalen Mobiltelefons auf die Telefon-Funktion, das Text-Messaging und minimale Organizer-Funktionen. Wobei der Trend dahin geht, in dem relativ kleinen Mobiltelefon soviel Funktionen wie möglich unterzubringen. Neben den sich daraus ergebenden Einsatzgebieten besitzen all diese Geräte noch weitere wesentliche Merkmale. Dazu gehören zum Beispiel der graphische Darstellungsbereich und das Eingabemedium, die aufgrund ihrer Gegebenheiten die verschiedensten Eingabe-Methoden erforderlich machen.

## 2 Problemstellung

PDA's und Smartphones werden immer populärer. Dies ist bei dem riesigen Funktionsumfang verständlich. Die Funktionspalette von Smartphones reicht vom Handy über MP3-Player, Video-Player, Office-Anwendungen, Digitalkamera, WLAN bis hin zum ausgereiften Navigationssystem. Außerdem sind sie viel billiger als Notebooks und wesentlich kleiner, so dass man sie problemlos überall hin mitnehmen kann. Ebenfalls verfügen Handys über immer mehr Organizer-Funktionen. Zudem bieten fast alle Hersteller Software für die Synchronisation mit Microsoft Outlook an.

Durch die immer komplexer werdenden Anwendungen und die Möglichkeit Office-Dokumente mit den mobilen Endgeräten zu betrachten, zu bearbeiten und zu erstellen, ergibt sich die Frage nach passenden Eingabemöglichkeiten. Die kleinen Alleskönner werden Notebooks in ihren Anwendungsmöglichkeiten immer ähnlicher, nur mit dem Nachteil, dass sie in der Regel keine QWERTZ-Tastatur besitzen. Dies erschwert die Texteingabe vor allem von größeren Textmengen bei z.B. Sitzungs-Mitschriften oder Gesprächsnotizen. Aber beispielsweise auch bei der Eingabe von einem gewünschten Zielort bei der Navigationssoftware, beim Ausfüllen von Formularen während des Surfens im Internet oder dem Schreiben von Emails oder SMS-Nachrichten, werden bedingt durch die alternativen Eingabegeräte besondere Eingabe-Techniken erforderlich.

In den folgenden Kapiteln werden diese verschiedenen Eingabe-Geräte und Methoden genauer erläutert, kategorisiert und bewertet.

### **3 Klassifikation von Eingabeverfahren**

Aus dem graphischen Darstellungsbereich und dem Eingabegerät ergibt sich schon die Klassifikation der Eingabe-Methoden.

Wo ein Notebook eine QWERTZ-Tastatur besitzt, die fast mit der des Desktop-Computers identisch ist, benötigt ein PDA bzw. ein Smartphone einen Stift (Stylus) zur Eingabe. Ein Mobiltelefon hingegen besitzt die traditionelle 12-Button-Tastatur. Das Notebook mit einer klassischen Tastatur, auch QWERTZ-Tastatur genannt, wird hier nicht weiter behandelt, da dafür keine besonderen Eingabeverfahren notwendig sind. Andere Tastaturarten und Forschungskeyboards, wie beispielsweise Chord-Keyboards, die meist aus 5 Tasten bestehen und die Buchstaben durch Kombinationen dieser 5 Tasten ausgewählt werden können, werden hierbei nicht weiter erläutert.

Somit ergibt sich eine Klassifikation in die 2 Kategorien der Stylus-basierten Eingabemethoden und der 12-Button-Eingabeverfahren.

Im Folgenden werden unter Punkt 3.1 die verschiedenen Stylus-basierten Verfahren vorgestellt. Dabei liegt das Hauptaugenmerk auf dem Verfahren T-Cube. Unter Punkt 3.2 werden dann die unterschiedlichen Button-Interaktionstechniken näher behandelt. Dabei liegt der Fokus auf den 12-Button-Eingabeverfahren, insbesondere auf dem Verfahren TiltText.

#### **3.1 Stylus-basierte Eingabemöglichkeiten**

Der momentane Trend geht dahin, dass mobile Endgeräte immer kleiner werden, aber über immer mehr Funktionen verfügen. Als problematisch erweist sich jedoch die Wahl des Eingabemediums. Während es bislang einige Implementierungen von Mini-Tastaturen gab, die sich jedoch für die Eingabe von viel Text als ungeeignet erwiesen haben [VN94], verfügen viele der kleinen Multitalente über einen Stift (Stylus) als Eingabemedium. Mittlerweile wurden auch schon mehrere Eingabemethoden entwickelt.

### 3.1.1 *Handschrifterkennung*

Weit verbreitet ist die Handschrift-Erkennung, die heutzutage in fast allen PDAs und vielen Smartphones integriert ist. Der Anwender schreibt mit dem Stift direkt auf dem Bildschirm. Dabei hat er entweder einen kleinen Bereich im unteren Teil des Bildschirms oder den kompletten Bildschirm zur Verfügung. Die technische Umsetzung und die Exaktheit der Schrifterkennung sind allerdings problematisch [KRS00]. Zum Beispiel müssen sehr viele Informationen wie der Strichverlauf und der Schreibdruck verarbeitet werden. Des Weiteren muss das Eingegebene zerlegt und anschließend interpretiert werden. Da die meisten mobilen Geräte noch nicht so leistungsstark sind, entsteht meist eine gewisse Verzögerung, bis der eingegebene Text interpretiert auf dem Bildschirm erscheint. Wichtiger ist die Treffergenauigkeit, da sie für die Benutzerakzeptanz und die Durchsetzung auf dem Markt ausschlaggebend ist. Man unterscheidet bei der Handschrifterkennung zwischen Wort- und Zeichenerkennung.

#### 3.1.1.1 Worterkennung

Die Worterkennung ist eine komplexe Handschrifterkennungsmethode [KRS00]. Dies führt oft zu Fehlern bei der Erkennung. Meistens muss man sich als Anwender für eine korrekte Erkennung um eine ordentliche und saubere Handschrift bemühen, da sonst die Trefferquote der Texterkennung sinkt. Als Beispielsoftware für die Worterkennung wäre zum Beispiel der Microsoft Transcriber zu nennen, wo man auf dem kompletten Bildschirm des PDAs schreiben kann [MS].

#### 3.1.1.2 Zeichenerkennung

Im Gegensatz zu der Worterkennung werden hierbei, wie der Name schon sagt, nur die einzelnen Zeichen erkannt. Somit reduziert sich der Rechenaufwand und die Trefferquote steigt. Dafür ist meist ein kleiner Eingabebereich im unteren Teil des Displays reserviert. Allerdings bringt diese Methode ein Problem mit sich. Und zwar ist es problematisch zu erkennen, wann die Buchstabeneingabe abgeschlossen ist, da für manche Buchstaben wie zum Beispiel das ‚t‘ mehrere Striche notwendig sind. Dies führte zu der Entwicklung neuer Methoden wie Unistroke und Graffiti [KRS00]. Diese beiden Methoden benutzen ein neues Alphabet, bei dem der Stift bei der Eingabe der Zeichen nicht mehr angehoben werden muss.

Die von Unistroke implementierten Symbole haben meist wenig Ähnlichkeit mit den normalen Buchstaben. Das Hauptaugenmerk lag dabei auf der Schlichtheit der Symbole, durch die der Erkennungsaufwand erheblich reduziert wird. So wurden zum Beispiel den sehr oft verwendeten Buchstaben wie ,e, a, t, i, r' einfache Linien in den orthogonalen und diagonalen Richtungen zugewiesen. Unistroke ist wenig benutzerfreundlich für Anfänger, aber es ermöglicht fortgeschrittenen Anwendern hohe Eingabegeschwindigkeiten. Abbildung 1a zeigt das Alphabet von Unistroke.

Das Alphabet von Graffiti, welches bei Palms Verwendung findet, hat viel mehr Ähnlichkeit mit dem normalen Alphabet. Dies erhöht jedoch wieder den Rechenaufwand bei der Erkennung, ist aber somit für Anfänger leichter zu erlernen. Die Zeichen vom Graffiti-Alphabet sind in Abbildung 1b zu sehen.



Abbildung 1: Alphabet von Unistroke (a) und Graffiti (b) [KRS00]

### 3.1.2 Soft-Keyboard

Ein weiteres Beispiel für die Stylus-basierte Eingabe ist das so genannte Soft-Keyboard. Dabei wird auf den Bildschirm des PDAs eine kleine Tastatur abgebildet (siehe Abbildung 2), so wie man sie vom herkömmlichen Computer her kennt.



Abbildung 2: Soft Keyboard des XDAII

Der User kann dann einfach schreiben, indem er mit dem Stift auf die Tasten drückt. Soft-Keyboards werden oft als Alternative zu der Handschrifterkennung genutzt, sind dem Nutzer optisch schon bekannt und daher einfach zu bedienen. Allerdings sind die Tasten des Soft-Keyboards recht klein und somit zum Beispiel im Auto schwer zu treffen, auch wenn man möglicherweise zwischen verschiedenen Größen wählen kann.

### 3.1.3 T-Cube

Jedoch sind alle diese Verfahren in der Eingabe-Schnelligkeit begrenzt. T-Cube ist ein schnelleres Verfahren für die Texteingabe [VN94]. Im Folgenden wird das Verfahren kurz beschrieben. Weiterhin wird erklärt, wie das Verfahren entstanden ist und getestet wurde, bevor anschließend eine Bewertung erfolgt.

T-Cube funktioniert wie folgt: Die Auswahl der Zeichen, Buchstaben und Ziffern erfolgt mit kleinen Handbewegungen bzw. Stichzügen. Eine solche Handbewegung lässt sich durch einen Startpunkt und seine Richtung definieren. Auf dem Bildschirm befindet sich ein kleines T-Cube-Fenster. In diesem Fenster wird ein Kreis angezeigt. Dieser Kreis wird als Target bezeichnet. Dieses Target besteht aus neun Zellen, wie es in Abbildung 3 dargestellt ist.

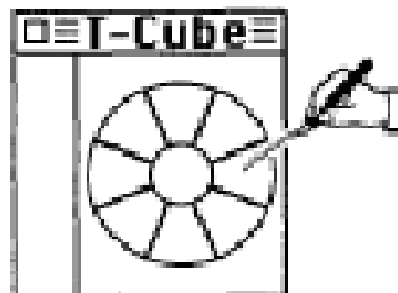


Abbildung 3: T-Cube Target [VN94]

Dabei gibt es eine zentrale und acht weitere Zellen, die ringsherum angeordnet sind. Der Durchmesser des Kreises entspricht in etwa 0,7 bis 1,7 cm. Drückt der Benutzer nun mit dem Stift in eine der 9 Zellen, öffnet sich nach nur einem Bruchteil einer Sekunde ein weiteres Menü-Fenster, welches als Pie-Menü bezeichnet wird. Es besteht aus 8 Zellen. Dies begründet sich darin, dass man seine Handbewegung vom Ausgangspunkt in 8 weitere Richtungen machen kann, nämlich horizontal,

vertikal, sowie in die 2 diagonalen Richtungen. Somit sind von jeder der 9 Ausgangszellen des Targets 8 weitere Zellen zu erreichen. Insgesamt sind das 72 Möglichkeiten, die mit einem Zeichen, Buchstaben, einer Zahl oder einer Funktion belegt werden können und jeweils mit einer bestimmten Handbewegung zu erreichen sind. Anders als bei herkömmlichen Menüs, wird diese nicht direkt unter der Stiftspitze angezeigt, sondern je nach dem ob der Anwender Links- oder Rechtshänder ist, kann sich der Benutzer das Pie-Menü rechts oder links vom Target anzeigen lassen. Anderfalls würde die Hand des Users das Pie-Menü schon halb verdecken.

Nachdem der Nutzer in eine Target-Zelle selektiert und sich das Pie-Menü geöffnet hat, braucht er nur noch eine Bewegung in die gewünschte Richtung machen. Dabei wird kurz nach Beginn der Bewegung und sobald die Richtung absehbar ist, das entsprechende Symbol hervorgehoben. Ebenfalls wird der Strichzug auf dem Display durch eine dicke Linie dargestellt, wie in Abbildung 4 zu sehen ist. Sobald der User den Stift wieder anhebt, wird beispielsweise der Buchstabe getippt.

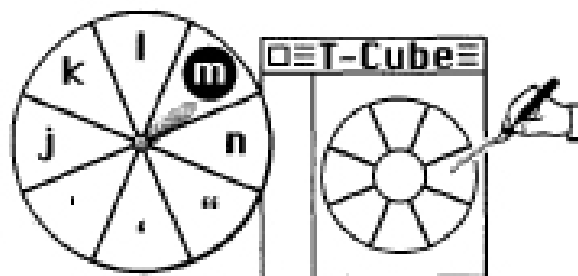
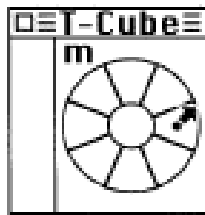


Abbildung 4: Target mit Pie-Menü [VN94]

T-Cube hat für Anfänger und Experten verschiedene Modi implementiert. So gibt es für T-Cube-Neulinge einen Trainingsmodus. In diesem Modus erscheint das Pie-Menü, sobald der Stift in der Nähe des Targets ist. Dadurch ist das Pie-Menü schon sichtbar, wenn der Stift über der Target-Zelle ist. So sieht der Anwender immer gleich den Inhalt der Zelle, da er diesen als Anfänger ja noch nicht kennt. Ist die Target-Zelle die richtige, so braucht der Benutzer nur noch mit dem Stift auf das Display drücken, ihn in die gewünschte Richtung bewegen und wieder anheben. Fortgeschrittene Nutzer können auch noch auf das Pie-Menü zurückgreifen falls dies erforderlich ist. Experten, die die Inhalte kennen, nutzen T-Cube ohne das Pie-Menü. So lassen sich für Experten sehr hohe Eingabegeschwindigkeiten erzielen [VN94].

T-Cube hat zusätzlich noch ein audio-visuelles Feedback implementiert. Dies soll dem Anfänger helfen die richtigen Strichzüge zu erlernen. So werden dem User bei dessen Eingabe kurz das gewählte Zeichen und die ideale Handbewegung angezeigt, siehe Abbildung 5. Ebenfalls werden verschiedene Töne wiedergegeben, die dem User eine korrekte bzw. eine ungenaue Bewegung deutlich machen sollen. So ertönt ein kurzer hoher Ton, wenn die Target-Zelle in der Mitte berührt wurde. Wird die Target-Zelle hingegen nah am Rand gedrückt, erklingt ein tieferer Ton. Ähnlich wird es dem Benutzer signalisiert, wenn er keine exakte orthogonale bzw. diagonale Bewegung gemacht hat.



*Abbildung 5: Feedback bei T-Cube [VN94]*

Die Zeichen wurden nicht willkürlich auf die 9 Target-Zellen verteilt, sondern ganz bewusst nach speziellen Kriterien angeordnet. Dafür wurde ein Experiment durchgeführt. Es mussten Probanden auf einem Display angezeigte Targets und Strichzüge nachzeichnen. Man erlangte dadurch Erkenntnisse über die Geschwindigkeit für jede einzelne Handbewegung und über die Zeiten zwischen einzelnen Handbewegungen. Detaillierter wird auf die Experimentdurchführungen nicht eingegangen. Nähere Informationen dazu sind in [VN94] zu finden. Die Ergebnisse der Studie sind in Abbildung 6 dargestellt. Daraus kann man die Geschwindigkeiten der verschiedenen Richtungen für jede der neun Zellen erkennen. Die graue Kreisfläche stellt dabei die Durchschnittsgeschwindigkeit der jeweiligen Zelle unabhängig von den verschiedenen Richtungen dar.

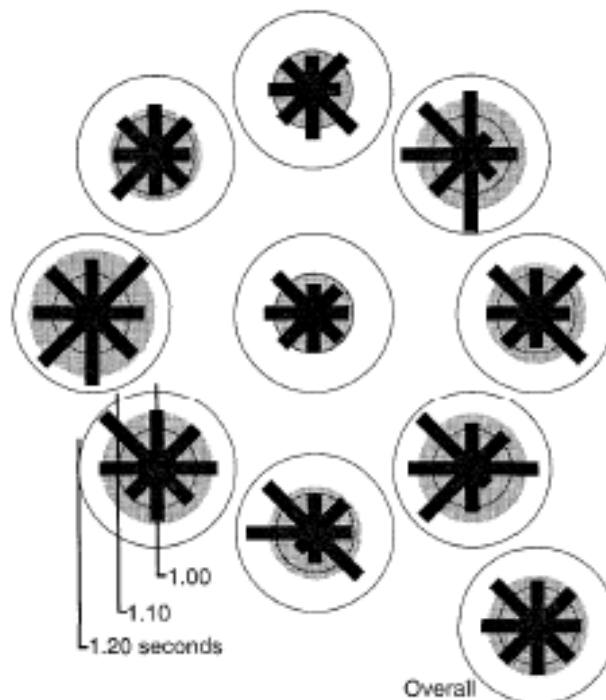


Abbildung 6: Ergebnisse des Geschwindigkeitsexperiments [VN94]

Demnach ist Zelle in der Mitte die schnellste, gefolgt von der nördlichen, nordwestlichen, südlichen und östlichen Zelle. Außerdem scheinen Bewegungen nach Westen insgesamt die langsamsten zu sein. Weiterhin scheint es keine generellen Geschwindigkeitsunterschiede zwischen orthogonalen und diagonalen Bewegungen zu geben. Diese Erkenntnisse beeinflussten die Anordnung der Inhalte der Menüs. So wurden häufig gewählte Zeichen den Zellen mit schnellen Bewegungen zugeordnet. Ein weiterer, wichtiger Aspekt ist die Clusterung von Zeichen, damit der Nutzer in einer gewissen Art vorhersehen bzw. erahnen kann, wo sich das gewünschte Zeichen befinden könnte. Laut [VN94] ist dabei die Zuordnung von Zeichen zu den Target-Zellen wichtiger als die Anordnung innerhalb des Pie-Menüs. Demnach gehören das Leerzeichen, Backspace und Shift zu den am meisten verwendeten Funktionen. Aus diesem Grund wurden diese so genannten Meta-Keys mit weiteren Funktionen auf die schnellste, also die mittlere Target-Zelle gelegt. Diese Meta-Keys fungieren als eine Art Schalter. Man kann sie durch die gleiche Aktion wieder deaktivieren. Nachdem ein Zeichen geschrieben wurde sind die Meta-Keys, bis auf Caps-Lock, wieder deaktiviert. Wurde ein Meta-Key aktiviert, so wird er im T-Cube-Fenster, links neben dem Target zur Kontrolle angezeigt.



Eine in [VN94] vorgestellte Langzeitstudie hat gezeigt, dass Anwender bereits nach kurzer Zeit mit dem Layout von T-Cube vertraut sind und ohne Pie-Menüs auskommen.

Neben der schneller Erlernbarkeit gegenüber Unistroke bietet T-Cube weitere Vorteile. Es beansprucht einen viel kleineren Bereich des Displays als Soft-Keyboards und Handwriting Recognition-Systeme, die spezielle Eingabebereiche auf dem Display reservieren. Zudem verfügt T-Cube über einen sehr großen Zeichensatz bei 72 möglichen Strichzügen. Außerdem ist T-Cube laut [VN94] durchaus konkurrenzfähig mit Unistroke und Soft-Keyboards. Unistroke besitzt jedoch einen großen Vorteil gegenüber Soft-Keyboards und T-Cube. So kann es nämlich ohne visuelle Kontrolle verwendet werden. Soft-Keyboards benötigen hingegen ein hohes Maß an visueller Aufmerksamkeit. Ob T-Cube auch ohne visuelle Kontrolle auskommt, muss noch genauer untersucht werden. Allerdings ermöglicht T-Cube eine Anwendung ohne die Hand zu bewegen. Zudem ist die Anwendung für Links- und Rechtshänder, wie bereits beschrieben, möglich.

Einen enormen Vorteil hat T-Cube dadurch, dass es sowohl durch Neulinge als auch für Experten nutzbar ist. So ist es für Neulinge mit dem Trainings-Modus leicht erlernbar und bietet Experten trotzdem die Möglichkeit hohe Geschwindigkeiten zu erreichen. Soft-Keyboards sind auch für Neulinge schnell anwendbar, bieten aber den Experten keine speziellen Möglichkeiten für schnellere Eingaben. Unistroke hingegen wenig Anfänger-freundlich, lässt aber bei Experten hohe Eingabegeschwindigkeiten zu.

### **3.2 12-Button-basierte Eingabemöglichkeiten**

Für Mobiltelefone hat sich die 12-Button-Tastatur als Standardeingabegerät durchgesetzt. Unter einer 12-Button-Tastatur versteht man eine Tastatur, die aus zehn Tasten für Zahlen und zwei Tasten für zusätzliche Symbole wie „\*“ und „#“ bestehen. Durch die begrenzte Anzahl von Tasten ist eine Mehrfachbelegung erforderlich. Dabei werden die Buchstaben den Tasten in alphabetischer Reihenfolge zugeordnet. Die ITU hat mit dem ITU E.161 Standard einen Standard für die

Belegung der 12-Button-Tastatur festgelegt [WB04]. Die Anordnung ist in Abbildung 9 dargestellt. Allerdings haben die meisten Mobiltelefone mittlerweile weitere Tasten mit speziellen Funktionen, wie z.B. einen Mini-Joystick oder eine Taste um in das Menü zu gelangen.



*Abbildung 9: Standard 12-Key Keypad nach ITU E161 [WB04]*

Im Jahr 2003 wurde ca. 500 Milliarden SMS weltweit verschickt [GSM]. Durch die enorme Zahl an versendeten SMS ist die Bestrebung nach schnellen Eingabemethoden sehr groß. In diesem Abschnitt werden mehrere dieser Methoden vorgestellt und gegenübergestellt. Die Gegenüberstellung bezieht sich meist auf Multitap, da dieses am weitesten verbreitet ist.

### *3.2.1 Multitap*

Multitap ist das traditionelle Eingabe-Verfahren bei Mobiltelefonen. Grundgedanke dieses Verfahrens ist der, dass eine Taste sooft gedrückt wird, bis der entsprechende Buchstabe erscheint [PS2]. So muss man z.B. die Taste ‚6‘ einmal für ein ‚m‘, zweimal für ein ‚n‘ und dreimal für ein ‚o‘ drücken. Dies ist eine sehr einfache Eingabe-Möglichkeit und für Anfänger schnell zu verstehen. Allerdings ist das Eingeben von Buchstaben derselben Taste umständlich. So muss man z.B. nach der Eingabe von einem ‚a‘ eine gewisse Zeit warten, bis man ein ‚b‘ eingeben kann. Man hat jedoch auch die Möglichkeit mit den Cursor-Tasten das Time-out zu überspringen. Dies erhöht zwar die Anzahl der Tastenbetätigungen, ermöglicht für geübte Nutzer aber schnellere Eingabe-Geschwindigkeiten.

### 3.2.2 Less-Tap

Bei der Entwicklung von Less-Tap wollte man ein Eingabe-Verfahren kreieren, das schneller als Multitap und kompatibel zu der Standard-Tastatur ist [PS1]. Außerdem sollte es einfach zu erlernen und einfach anzuwenden sein. Bei Less-Tap ist die Zuordnung von Buchstaben zu den jeweiligen Tasten identisch mit der bei Multitap. Lediglich die Reihenfolge unter den Buchstaben pro Taste differiert. Hintergrund dabei ist, die am häufigsten verwendeten Buchstaben mit nur einem Tastendruck, die zweit-häufigsten Buchstaben mit zwei Tastenbetätigungen usw. zu erreichen. Jedoch ist die Buchstaben-Häufigkeitsverteilung sprachabhängig. Die in Abbildung 10 dargestellte Belegung ist die Belegung für die englische Sprache.

1	2acb	3edf
4ihg	5lkj	6onm
7srpq	8tuv	9wxyz
*	0	#

Abbildung 10: Tastaturbelegung von Less-Tap [PS1]

Ein in [PS1] durchgeführtes Experiment lieferte folgende Ergebnisse. Wie Abbildung 11 verdeutlicht, konnten bei Less-Tap ca. 65% aller Buchstaben eines einzugebenden Textes mit einem einzigen Tastendruck eingetippt werden. Bei Multitap hingegen traf dies nur für weniger als 50% der Buchstaben zu. Während die Anzahl der Buchstaben die 2 Tastenbetätigungen benötigen bei den zwei Verfahren in etwa gleich sind, ist diese bei Less-Tap für Buchstaben mit drei und vier Tastenbetätigungen deutlich geringer.

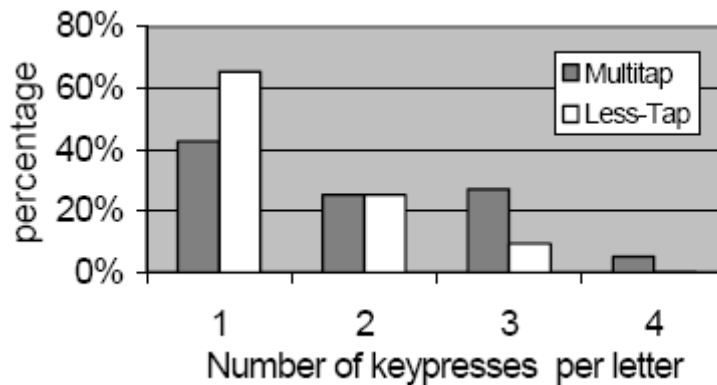


Abbildung 11: Tastenbetätigungen pro Buchstabe bei Lesstap [PS1]

Allerdings ist anzumerken, dass Less-Tap im Test bei der Eingabe weniger Tastenbetätigungen pro Sekunde (KSPS) erreichte. Dies kann damit begründet sein, dass die Tastenbelegung ‚ACB‘ ungewohnter ist als das vertraute ‚ABC‘ und somit weniger intuitiv verwendet wird. Dies widerspricht jedoch der geringeren Fehlerrate von Less-Tap gegenüber Multitap. Jedoch wurden mit Less-Tap mehr Wörter pro Minute (WPM) geschrieben, was dadurch begründet ist, dass Less-Tap weniger Tastenbetätigungen pro Zeichen (KSPC) benötigt.

### 3.2.3 Two-Key

Bei diesem Verfahren benötigt der User, wie der Name schon erahnen lässt, zwei Tastenbetätigungen pro Buchstabe. Dabei wählt der User mit dem ersten Tastendruck die Gruppe der Buchstaben aus. Also wird beispielsweise mit der Taste ‚2‘ die Buchstabengruppe ‚ABC‘ ausgewählt. Mit dem zweiten Tastendruck wählt der Benutzer schließlich den endgültigen Buchstaben aus dieser Gruppe aus. So wird z.B. durch zweimaliges Drücken der Taste ‚2‘ der Buchstabe ‚b‘ gewählt. Durch diese Verfahrensweise entfällt das Problem des Wartens bei aufeinander folgenden Buchstaben der gleichen Taste. Es werden so höhere Eingabegeschwindigkeiten als bei Multitap erreicht. Allerdings hat sich dieses Verfahren nicht verbreitet [PS2].

### 3.2.4 Wörterbuch-basierte Verfahren

Verschiedene Firmen haben solche Wörterbuch-basierten Verfahren bereits entwickelt. Dazu gehören zum Beispiel iTap von Motorola, eZitext von Zi Corp und

T9 von Tegic Communications. Dabei ist T9 von Tegic Communications [TEG] das bekannteste und am weitesten verbreitete Wörterbuch-basierte Verfahren. Allerdings ist deren Funktionsweise sehr ähnlich.

Der Benutzer braucht für jeden Buchstaben nur eine Taste drücken. Man braucht also für ein Wort mit 3 Buchstaben nur 3 Tasten drücken. Das System durchsucht dann das Wörterbuch für jede mögliche Permutation dieser Tastenfolge nach passenden Wörtern. Bei mehreren Treffern wird das am meisten wahrscheinlichste Wort angezeigt. Sollte das Wort nicht das gewünschte sein, kann der Nutzer zwischen den gefundenen Wörtern manuell wechseln, bis das gewünschte auf dem Display erscheint. Die Schnelligkeit des Verfahrens ist davon abhängig, ob das gesuchte Wort im Wörterbuch enthalten ist und ob es korrekt buchstabiert wurde. Sollte das gesuchte Wort nicht im Wörterbuch enthalten sein, hat man meist die Möglichkeit es per Multitap erneut einzugeben und dem Wörterbuch hinzuzufügen.

T9 hat einen KSPC nahe an 1, aber nur falls dieses im Wörterbuch enthalten ist. Außerdem ist dieses Verfahren weniger geeignet für umgangssprachliche Wörter, die nicht im Wörterbuch enthalten sind. Allerdings sind es ja überwiegend Jugendliche, die SMS versenden und diese drücken sich meistens umgangssprachlich aus. Weiterhin wird für jede angebotene Sprache ein eigenes Wörterbuch benötigt, was speicherintensiv ist.

### *3.2.5 Präfix-basierte Verfahren*

LetterWise ist ein Verfahren, das Präfix-basiert arbeitet. Das heißt es arbeitet nicht mit einem Wörterbuch, sondern mit der Wahrscheinlichkeit von Buchstabenfolgen bzw. Präfixen. Diese Wahrscheinlichkeiten sind in einer Datenbank abgelegt. Bei diesem Verfahren wird der folgende Buchstabe vorhergesagt, ausgehend von den bisher geschriebenen Buchstaben des Wortes. Wurde also beispielsweise bisher ‚aut‘ geschrieben und anschließend die Taste ‚6‘ gedrückt, erscheint ein ‚o‘, da in der deutschen Sprache ‚auto‘ mehr wahrscheinlich ist als ‚autm‘ oder ‚autn‘. Wird der falsche Buchstabe vorhergesagt, kann der User über eine Weiter-Taste zu dem

Buchstaben wechseln, der am zweit-wahrscheinlichsten ist und so weiter. Die Wahrscheinlichkeit der korrekten Vorhersage steigt mit der Länge des Präfixes. Eine in [MKS1] beschriebene Studie zeigt, dass 50,1% aller Wörter ohne ein einziges Drücken der Next-Taste eingegeben werden können. Bei den restlichen 49,9%, bei denen die Next-Taste gedrückt werden muss, ist dies beim ersten Buchstaben nötig. Nachdem der erste Buchstabe des Wortes korrekt eingegeben wurde, wird das Drücken der Weiter-Taste immer unwahrscheinlicher, wie Abbildung 12 zeigt.

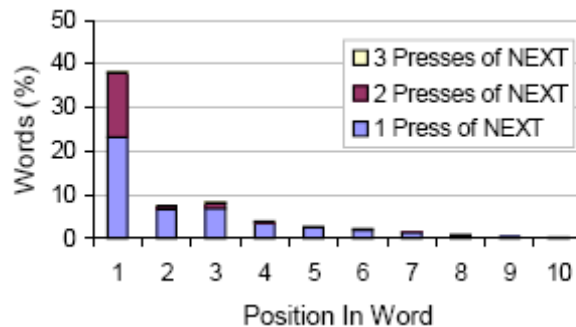


Abbildung 12: Betätigung der Next-Taste bei LetterWise [MKS1]

Dieses Verfahren benötigt weitaus weniger Speicherplatz für die Datenbank als T9 für das Wörterbuch. Außerdem lässt es anders als T9 auch die Eingabe von Wörtern, die nicht zum normalen Wortschatz gehören oder die Eingabe von Abkürzungen zu, ohne in einen speziellen Lern-Modus wechseln zu müssen. Selbst ausländische Wörter können problemlos eingegeben werden. Dies erfordert maximal ein paar zusätzliche Betätigungen der Next-Taste. Das in [MKS1] vorgestellte Experiment ergab, dass LetterWise sogar 36% schneller in der Eingabe ist als Multitap.

### 3.2.6 TiltText

Dieses Verfahren versucht die Mehrfachbelegung durch die Aufteilung der 26 Buchstaben auf die Tasten 2 bis 9 mittels Ausrichtung des Handys bei der Eingabe zu überwinden. Dafür wird neben der 12-Button-Tastatur noch ein Neigungs-Sensor im Handy verwendet. So wird beim Drücken einer Taste und gleichzeitigem Neigen des Handys in eine von vier Richtungen (links, rechts, vor, zurück) der jeweilige Buchstabe gewählt. Wie Abbildung 13 verdeutlicht, gibt es dabei folgende Zuordnungen der Neigungsrichtungen zu den Buchstaben. Beim Drücken der Taste

ohne es zu neigen wird die Zahl gewählt. Beim Neigen nach links wird der erste Buchstabe, beim Neigen nach vorn der zweite, beim Neigen nach rechts der dritte gewählt. Wird das Handy nach hinten geneigt wird der vierte Buchstabe gewählt, sofern er vorhanden ist.



Abbildung 13: TiltText [WB03]

TiltText lehnt sich stark an TiltType an. TiltType wurde für sehr kleine mobile Geräte entwickelt. Hierbei wird es nur kurz beschrieben, um die Ähnlichkeit zu TiltText darzustellen. Die kleinen Geräte mit TiltType konnten sogar am Handgelenk getragen werden. Die Eingabe bei diesem Verfahren erfolgte mit lediglich 4 Tasten kombiniert mit dem Neigen des Geräts. Dabei wurde zwischen acht verschiedenen Richtungen unterschieden, wie Abbildung 14 zeigt.

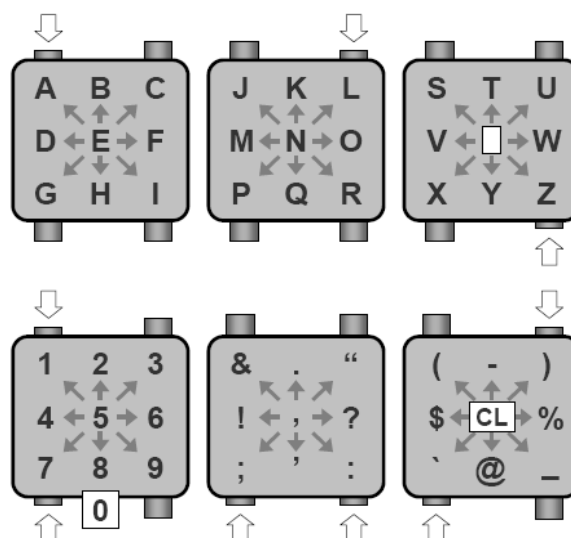


Abbildung 14: TiltType [PCS1]

Im Gegensatz zu TiltType verwendet TiltText die Standard-12-Button-Tastatur, was die Realisierbarkeit und Durchsetzung des Systems erhöhen sollte. Außerdem werden statt acht Richtungen lediglich vier verwendet. Dies reduziert die Genauigkeit der Neigungsbewegung für den Nutzer.

Für die Einführung der Groß- und Kleinschreibung werden zwei verschiedenen Varianten vorgeschlagen. Einerseits könnte man dies durch eine Extra-Taste realisieren, andererseits durch das Ausmaß der Neigung, wie Abbildung 15 verdeutlicht. Dies würde aber eine Nutzung ohne visuelle Kontrolle erschweren.

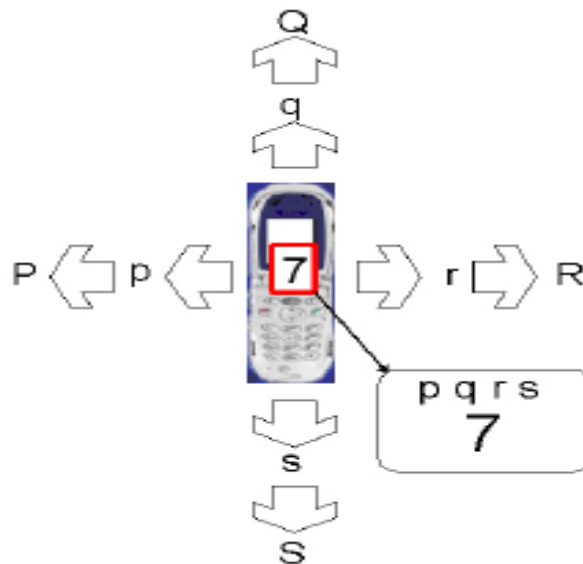


Abbildung 15: Groß- und Kleinschreibung bei TiltText [WB03]

Für TiltText wurden mehrere Algorithmen entwickelt. So gibt es zum Beispiel Key Tilt, Absolut Tilt und Relative Tilt. Bei TiltText mit dem Key Tilt-Algorithmus werden die Neigungswerte zwischen Drücken und Loslassen der Taste ermittelt. Der Benutzer muss also zuerst die Taste drücken, dann das Handy neigen und schließlich die Taste wieder loslassen. In einem von [WB03] durchgeführten Experiment war diese Variante jedoch viel langsamer als Multitap.

Beim Absolut Tilt-Algorithmus wird der Neigungswert in Relation zu einem festen, absoluten Wert ermittelt. Der User braucht dabei also nur das Handy neigen und die Taste drücken. Allerdings lässt der Benutzer während der Text-Eingabe in der Realität selten seine Arm-Position gleich. Deshalb muss der Nutzer nach einer Änderung der Arm-Stellung die absolute Position des Telefons mit einer bestimmten

Taste neu festlegen. Allerdings muss das Mobiltelefon bei der Eingabe von zwei Zeichen entgegen gesetzter Richtung, wie beispielsweise ‚a‘ und ‚c‘, zuerst nach links, dann wieder gerade und schließlich nach rechts geneigt werden. Das ist eine Strecke mehr als bei Key Tilt notwendig ist. Jedoch hat dieser Algorithmus auch einen Vorteil gegenüber Key Tilt. Bei der Eingabe von Buchstaben, die dieselbe Richtung haben, kann das Telefon in der Stellung verbleiben und die jeweiligen Buchstaben eingegeben werden.

Der Relative Tilt-Algorithmus nutzt einen variierenden Ursprungs-Wert. Dieser wird beim Beginn der Bewegung gesetzt. Um zu entscheiden können, wann eine Bewegung beginnt, wird permanent auf eine Änderung der Handy-Ausrichtung oder der Neige-Bewegung geachtet. Somit gibt es keinen absoluten Wert, der bei einer Änderung der Arm-Position zurückgesetzt werden muss. Außerdem braucht man das Handy bei der Eingabe von Buchstaben unterschiedlicher Richtung nicht zuerst wieder in den Ursprung zu bewegen. Der Endpunkt der ersten Bewegung ist gleichzeitig der Anfangspunkt der zweiten Bewegung.

Bei einem Experiment von [WB03] wurde der Absolut Tilt-Algorithmus verwendet. Dabei beklagten sich die Probanden lediglich über Daumenschmerzen, nicht aber über Schmerzen im Arm oder im Handgelenk, wie vielleicht zu erwarten gewesen wäre. Bei der Texteingabe-Geschwindigkeit war TiltText im Durchschnitt 16,3% schneller als Multitap. Nach [WB03] sind mit Relative Tilt noch bessere Werte zu erreichen, so dass TiltText mit den linguistischen Verfahren konkurrieren kann.

Der Vorteil von TiltText ist der, dass man für jeden Buchstaben lediglich einen Tastendruck benötigt. So lassen sich hohe Eingabegeschwindigkeiten erreichen. Ein anderer Vorteil ist der, dass die Eingabe von Text für geübte Nutzer ohne Hinschauen erfolgen kann.

### *3.2.7 Bewertung*

Nach einer Studie aus [MKS1] sind mit den Datenbank-basierten Eingabeverfahren weitaus höhere Eingabe-Geschwindigkeiten zu erreichen als mit Multitap, dem bisher am weitesten verbreiteten Eingabe-Verfahren. Zwischen beiden Datenbank-

basierten Verfahren ist demnach T9 mit ca. 40 Wörtern pro Minute (mit dem Daumen) am schnellsten.

Predicted Asymptotic Text Entry Rates (wpm)		
Method	Index finger	Thumb
<i>Multitap</i>		
- wait for timeout	22.5	20.8
- timeout kill	27.2	24.5
<i>T9</i>	45.7 <sup>a</sup>	40.6 <sup>a</sup>
<i>LetterWise</i>	38.1	33.7

Abbildung 16: Vergleich Eingabeverfahren [MKS1]

Laut [PS1] gelten folgende KSPC-Werte für die einzelnen Verfahren. Am schnellsten ist T9 mit 1.0072 KSPC, gefolgt von LetterWise mit 1.1500, Less-Tap mit 1.5266 und Multitap mit 2.0342 (die beiden letzten mit Umgehung der Wartezeit).

T9 und Letterwise sind zwar die schnellsten Verfahren, jedoch stellte MacKenzie [MKS1] fest, dass bei SMS in einer „eigene Sprache“ geschrieben wird. So werden oft eigene Abkürzungen und Kombinationen von Buchstaben und Zahlen verwendet, wie zum Beispiel 'b4' als Abkürzung für ‚before‘. Dafür sind T9 und LetterWise nicht geeignet. Für seltene, nicht im Wörterbuch enthaltene Wörter, erhöht sich die Eingabezeit bei T9 drastisch, was bei LetterWise nicht in dem Ausmaß der Fall ist.

So schnell T9 und LetterWise auch sein mögen, so besitzen sich doch einen großen Nachteil. Während Two-Key, Less-Tap und Multitap bei geübten Nutzern ohne jegliche visuelle Kontrolle auskommen, ist bei LetterWise und T9 ein Hinschauen zwingend erforderlich.

### 3.3 Weitere Eingabemöglichkeiten

Neben den zwei großen Kategorien der Stylus-basierten und den 12-Button-Eingabeverfahren, die den Markt beherrschen, gibt es noch weitere Interaktionsmöglichkeiten. Hierbei werden jedoch nur einige wenige Verfahren zur Vollständigkeit kurz vorgestellt.

#### 3.3.1 Ansteckbare Tastaturen

Heute schon weit verbreitet für die Eingabe von größeren Textmengen mit PDAs sind die ansteckbaren QWERTZ-Tastaturen. Diese haben den Vorteil, dass der User

damit bereits vertraut ist und so hohe Eingabe-Geschwindigkeiten erreicht werden können. Außerdem sind diese meist zusammenklappbar und benötigen so wenig Platz. Neben ansteckbaren Tastaturen gibt es mittlerweile auch Tastaturen die drahtlos über Bluetooth oder Infrarot mit dem mobilen Gerät verbunden sind.

### 3.3.2 *Optische Keyboard-Simulation*

Dieses Verfahren ist relativ neu und noch in der Entwicklung. Siemens hat solch ein System entwickelt. Dabei wird Tastatur mit einem Halbleiterlaser auf die Tisch-Oberfläche projiziert [SIE]. Die Eingabe erfolgt, wie Abbildung 17 verdeutlicht, mit einem speziellen Stift. Dieser leitet die Daten mittels Bluetooth an das mobile Gerät weiter. Allerdings wird es noch einige Zeit in Anspruch nehmen, bis diese Technik serienmäßig vertrieben wird.



Abbildung 17: Virtuelle Tastatur [SIE]

## 4 Zusammenfassung

In dieser Ausarbeitung wurde zuerst eine Klassifikation von Eingabeverfahren von mobilen Endgeräten getroffen. Dabei lag das Hauptaugenmerk auf Stylus-basierten und 12-Button-Eingabeverfahren. Anschließend wurden einige Methoden dieser Klassen vorgestellt und beschrieben. Bei den Stylus-basierten Eingabemethoden wurden Verfahren wie Soft-Keyboards und Handschrifterkennung nur kurz erwähnt. Lediglich auf T-Cube wurde näher eingegangen. Bei den 12-Button-Eingabeverfahren wurden Multitap, Less-Tap, Two-Key, T9, LetterWise und Tilttext näher erläutert. Dabei wurden all diese Verfahren verglichen und bezüglich ihrer Eingabeschwindigkeit bewertet. Für die Zukunft ist zu sagen, dass aufgrund des hohen SMS-Aufkommens immer Bedarf an schnelleren Eingabemethoden bestehen wird. Damit diese aber überhaupt eine Chance auf dem Markt haben und sich gegen die marktbeherrschenden Verfahren wie T9 oder Multitap durchsetzen, müssen sie vor allem einfach, schnell zu erlernen und ähnlich zu den bereits etablierten Techniken sein.

## Literaturverzeichnis

- [BAM] Website: <http://www.seda.wiai.uni-bamberg.de> (17.05.2005)
- [FIT] Website : <http://cartoon.iguw.tuwien.ac.at> (25.05.2005)
- [FKP1] Fitzmaurice, George; Khan, Azam; Pieké, Robert; Buxton, Bill; Kurtenbach, Gordon: Tracking Menus.  
In: Proceedings of the 16th annual ACM symposium on User interface software and technology (2003), Vancouver/Canada, ACM Press, S.71 – 79, ISBN: 1-58113-636-6.
- [FPC1] Fishkin, Kenneth P.; Partridge, Kurt; Chatterjee, Saurav:  
Wireless User Interface Components for Personal Area Networks.  
IEEE Pervasive Computing, vol. 01, no. 4, S. 49-55, 2002.
- [GSM] Website: <http://www.gsmworld.com> (22.06.05)
- [ISO] Poika Isokoski. A minimal device-independent text input method.  
Report A-1999-14, Department of Computer Science, FIN-33014,  
University of Tampere, Finland, 1999.
- [MKS1] MacKenzie, I. Scott; Kober, Hedy; Smith, Derek; Jones, Terry; Skepner, Eugene: LetterWise: Prefix-based Disambiguation for Mobile Text Input.  
In: Proceedings of the 14th annual ACM symposium on User interface software and technology (2001), Orlando/Florida, ACM Press, S. 111 – 120, ISBN:1-58113-438-X.
- [MS] Website: Mobile Solutions, Mobile Applications, and HandheldDevicesfrom Microsoft Windows Mobile.  
<http://www.microsoft.com/windowsmobile> (22.06.05)

- [NIE04] Niels, Ralph: Dynamic Time Warping: An intuitive way of handwriting recognition?  
Master Thesis, Radboud University Nijmegen, December 2004.
- [KRS00] Kahmann, Verena; Ruggaber, Rainer; Seitz, Jochen; Wolf, L.  
Seminar: Architektur vernetzter Systeme – Drahtlose Netze.  
<http://www.ubka.uni-karlsruhe.de/vvv/ira/2001/2/2.pdf> (24.05.2005).  
Wintersemester 2000/2001.
- [PCS1] Partridge, Kurt; Chatterjee, Saurav; Sazawal, Vibha; Borriello, Gaetano; Want, Roy: TiltType: AccelerometerSupported Text Entry for Very Small Devices. In: Proceedings of the 15th annual ACM symposium on User interface software and technology (2002). Paris/France: ACM Press, S. 201-204, ISBN: 1-58113-488-6.
- [PS1] Pavlovych Andriy; Stuerzlinger, Wolfgang:  
Less-Tap: A Fast and Easy-to-learn Text Input Technique for Phones.  
Graphics Interface 2003, Eds. Moller, Ware, AK Peters, Kuni 2003, S. 97-104, ISBN 1-56881-207-8.
- [PS2] Pavlovych Andriy; Stuerzlinger, Wolfgang:  
Model for non-Expert Text Entry Speed on 12-Button Phone Keypads.  
In: Proceedings of the CHI '04: Proceedings of the 2004 conference on Human factors in computing systems (2004): ACM Press, S.351-358, 2004.
- [SIE] Wbsite: Siemens: Handy mit virtueller Tastatur.  
<http://www.inside-handy.de/news/2887.html> (19.05.2005)
- [TEG] Website: <http://www.tegic.com> (22.06.05)
- [VN94] Venolia, Dan; Neiberg, Forrest: T-Cube: A Fast, Self-Disclosing Pen-Based Alphabet. In: Proceedings of the CHI '94: SIGCHI conference on

Human factors in computing systems (1994): ACM Press, S.265-270, 1994.

[WB03] Wigdor, Daniel; Balakrishnan, Ravin: TiltText: Using Tilt for Text Input to Mobile Phones. In: Proceedings of the UIST (2003), S.81-90, 2003.

[WB04] Wigdor, Daniel; Balakrishnan, Ravin: A Comparison of Consecutive and Concurrent Input Text Entry Techniques for Mobile Phones. In: Proceedings of the SIGCHI 2004: Conference on Human Factors in Computing Systems (2004): ACM Press, S. 81 – 88, 2004.

[ZK1] Zhai, Shumin; Kristensson, Per-Ola: Shorthand Writing on Stylus Keyboard. In: Proceedings of the CHI '03: Proceedings of the conference on Human factors in computing systems (2003): ACM Press, S.97-104, 2003.