

## ABSTRACT

Input to small devices is becoming an increasingly crucial factor in development for the ever-more powerful embedded market. Speech input promises to become a feasible alternative to tiny keypads, yet its limited reliability, robustness, and flexibility render it unsuitable for certain tasks and/or environments. Various attempts have been made to provide the common keyboard metaphor without the physical keyboard, to build “virtual keyboards”. This promises to leverage our familiarity with the device without incurring the constraints of the bulky physics. This research surveys technologies for alphanumeric input devices and methods with a strong focus on touch-typing. We analyze the characteristics of the keyboard modality and show how they contribute to making it a necessary complement to speech recognition rather than a competitor.

WWW.VTUCS.COM

# INTRODUCTION

Touch-typing or machine writing was invented for mechanical typewriters which had the current QWERTY key layout since 1874. While this interface is come to age, it survived because of its many positive aspects. Yet it is not feasible for the ever-smaller computing devices that house ever-more advanced functionalities. New alphanumeric interfaces include numeric keypads augmented with letters as on the cell phone and the Graffiti handwriting characters.

We survey the state of the art in alphanumeric input interfaces. After an overview of the related work in the area in section 2, we lay out the general space of interfaces for text input in section 3. In the following section we discuss characteristics and human factors of touch-typing in particular. Section 5 explains the criteria we examined with. Section 6 introduces and compares various touch-typing input methods and devices. We conclude with stating that keyboards – whether virtual or real – are very well suited to the task of alphanumeric input.

# ALPHANUMERIC INPUT

Since language and its manifestation in sentences, words, and letters is the human's primary means of communication, it has been researched extensively for the human-to-computer interaction as well. In this section, we take a look at three main categories: Speech recognition, handwriting recognition and sign language. Touch-typing is covered in section 4. Discussed are characteristics and shortcomings.

## Speech Recognition

Probably the most hailed UI, speech recognition (SR) is now at a stage where it can be successfully deployed in limited domains, such as in call centers for customer service or for voice dialing on mobile phones. Under these conditions it provides a highly user friendly, unobtrusive, flexible and efficient interface method. But careful estimates suggest that 100% recognition rates for large vocabularies will not be possible in the near future. Noisy environments, speaker particularities (accents, speech impediments) worsen the situation. One of the strongest arguments however for why we should not solely rely on SR lies in how our brain processes speech generation. As Shneiderman summarizes in [26], producing sounds that make up words and sentences occupies parts of the brain that are also used for general problem solving. This is contrary to how for example body movements are processed: Physical coordination does not conflict with problem solving. In other words, it is harder to think when the thought also has to be spoken aloud. Overall, even the versatile SR interface has its limitations.

## Handwriting Recognition

Handwriting recognition is another interface that offers itself to HCI due to its widespread mastery. It is equally hard as SR, especially connected-cursive handwriting. This is the reason for slightly modified alphabets like Graffiti [2] or entirely new penbased scripting languages like Cirrin [16] or Quikwriting [19] that are better suited for recognition by a computer. They achieve good recognition rates and can be implemented on devices with limited capabilities. Just as in writing on plain paper though, the throughput is limited because a single

actuator (the pen led by the fingers) is performing the communication task. One unit of information, a character, consists of a combination of straight and/or curved lines and/or dots. The amount of information a human can output through the medium of a pen is limited by motor skills rather than by cognitive skills. Furthermore, the improvement that can be achieved by training of motor capabilities peaks out much quicker than it does for cognitive capabilities. Stenography remedies this shortcoming of a single-pen interface by coding of language into more, and more complex, symbols. Yet stenography requires a significant amount of training, too much for the common user (see [3] for an explanation of the difficulties).

## **Sign Language**

Sign languages (SL) also code language into more symbols than there are letters in the English alphabet, thus a higher bandwidth can be achieved with fewer atomic elements. “Speaking” SL naturally does not involve the vocal system, although mouth movements frequently accompany hand and arm gestures. But – without proof – we guess that, contrary to speech processing, gesturing SL originates in parts of the brain that do not conflict with sentence forming. Therefore, it is conceivable that a SL-like modality can be developed that remedies the problems of SR and handwriting interfaces. Yet the drawbacks of using SL as input modality that must be overcome are big: The skill penetration in the population is small, extensive movements are required to communicate, signals are highly context dependent, and last but not least there are many difficulties associated with computer recognition of sign language.

# TYPING AS INPUT MODALITY

As derived in the previous section, no alternative input modality is the silver bullet to the UI problem. In this section, we show that touch-typing, while certainly not “perfect” either, does not have many of the drawbacks of other methods.

## Touch-Typing: Definition

We define touch-typing as any input method that employs discrete sensors, or sensed areas, or buttons, for one or a set of atomic symbols (letters, digits, or characters) of a language. Examples are the common keyboard, the keypad of a mobile phone, and onscreen keyboards on PDAs. This definition explicitly includes “virtual” buttons that only differ from the surrounding physique in that their extent is sensed by some technique for touch by a finger or pointer. We use the term keyboard, or keyboard metaphor, interchangeably for touch-typing interfaces.

## Characteristics and Human Performance

Touch-typing was born with the invention of the mechanical typewriter, and the common QWERTY layout followed us since 1874. Its greatest benefit is that all ten fingers can be used to operate it in very rapid sequential order. The interface is not restricted to communicating through a single pen. A seemingly related yet orthogonal benefit derives from the fact that each button generates only one bit of information. So in a sense, a keyboard is an input device with one degree of freedom (1 DOF), a pen or a mouse has 2 DOF (a 2-dimensional surface), and SL operates in 4 DOF (3-dimensional space plus time).

An exception which we will say to have 1.5 DOF (time being the extra half dimension) are buttons that can generate more than one character by pressing them repeatedly, for example those of mobile phone keypads. These keyboard types severely limit the achievable throughput, although a study [20] has shown that both motor and cognitive skills adapt to the common use with only the two thumbs. Shannon’s statistical methods and prediction [25], frequently called word disambiguation [4] and used for example in the T9 algorithm [28], can be employed to achieve bandwidths of up to 46 wpm [27]. (A more recent theoretical approach [14] suggests that

raw typing bandwidths up to 60 wpm are possible.) Another way to increase the amount of information each key can produce is to use complex moded or combinatorial operation, called chording keyboards. Individual key strokes do not generate a character, but only combinations of keys pressed simultaneously. Since  $n$  keys map to  $n+m$  symbols, chording keyboards also have a higher DOF than one. The maximum addressable space is  $2^n - 1$  symbols (less one because at least one key has to be pressed to recognize an action) with  $n$  keys.

Similar techniques as word disambiguation can be employed to even relax the requirement on the strict sequential ordering of key presses by performing context-sensitive reordering. This is frequently implemented in word processors that for example correct “hte” to “the”. This of course comes at the expense of greater context sensitivity of the input method.

Experimental observations suggest that the speed of classic touch typing with ten fingers is limited by cognitive skills rather than by the motor skills required to independently and rapidly move the ten fingers. In other words, the bottleneck for touch-typing bandwidth is the brain. For pen-based interfaces it is the dexterousness of the fingers rather than cognitive skills. This is good news: The brain can learn much quicker and better how to control body parts than the human physique takes to adapt to new requirements.

**Table 1. Bandwidth comparison of different UI methods for communicating alphanumeric data. All values are commonly used quantities, aside from the one for sign language which we guessed equal to reading prose, based on the ability to simulcast American Sign Language for live television.**

UI method	Bandwidth in words-per-minute
Conversation	280
Reading (prose)	250
Handwriting (paper)	30
Graffiti	20
Sign language	250
Keyboard touch-typing (average)	50
Keyboard t-t. (professional)	150

Another advantage of the keyboard metaphor over alternative methods of text entry is that it supports novice and expert users equally well. Keyboards afford both hunt and peck typing, which is advantageous if the key locations are not memorized, as well as touch-typing with ten fingers for advanced users.

## **The Influence of the Key(board) Layout**

The QWERTY key layout (i.e. which key maps to which language symbol) is not optimal in a number of ways. Various fixed-layout keyboards optimize the key arrangement for frequent alternating use of the left and right hand, short travel distances of fingers, decreasing load from index finger to pinky to compensate the decreasing strengths of these fingers and so forth. Most frequently cited is the Dvorak keyboard, whose key layout supposedly is far superior to QWERTY. Yet after a closer look [18][13][17], Dvorak layout is only marginally better in terms of maximum typing speed of experienced users. It does, however, fare superior in that it is easier and quicker to learn. Other approaches are more promising overall, for example Zhai et al [31] analytically predict the performance of various layouts and devise one with 40% predicted asymptotic performance improvement over QWERTY. We will not delve into further discussion here.

To mention are “soft” keyboards which (can) dynamically change the key arrangement for different users, situations, or even from keystroke to keystroke. Often times they also provide keys that produce a sequence of characters per key press. This benefit is of particular interest to virtual keyboards as they might combine a projection method with the input detection technique, thereby alleviating the need to switch the focus of attention from display device to a physically separate input device.

Keyboard layout modifications (as opposed to key layout which refers to changing the key-to-character mapping) such as the split keyboard on the other hand have been shown to have a very consistent impact. First proposed by Klockenberg et al in 1926 [12], split keyboards reduce the physical stress on hands and fingers for all users, novice and experienced.

## Virtual Keyboards

We define a virtual keyboard as a touch-typing device that does not have a physical manifestation of the sensing areas. That is, the sensing area which acts as a button is not per se a button but instead is programmed to act as one. So a sensing area could for example be realized with photo-electric sensors, active finger tracking methods, or a touch pad. The latter is different from a key pad as it does not have a-priori designated areas for buttons. Virtual keyboards that employ discrete sensing areas for each symbol (rather than chording methods) inherently allow for realization of a soft keyboard.

www.vtuics.com



# METRICS AND CHARACTERISTICS

We examined each of the methods and devices, called “virtual keyboards”, VK, for a number of characteristics which we consider relevant to a user interface. To simplify the comparison and future discussion of UIs we introduce a taxonomy specific to user input interfaces (and VKs in particular).

**Method or device:** Since we did not limit the survey to actual devices, we have to distinguish between mere suggestions of character-producing methods without implementations and actual operating devices. A third alternative are devices that could potentially function as a virtual keyboard, but it has not been used yet as such.

**Gear:** What devices are employed to realize the VK? Are there alternative methods that could shrink equipment size but can still produce the same VK method to the user? We also distinguish two main types of incarnations:

- a) Sensing areas are on a flat surface, e.g. a table: A regular key layout results in high familiarity with the interface method. Finger impacts with the surface produce tactile feedback. These VKs also allow for simple ways to visualize the keyboard.
- b) Sensing areas are on the user’s hand or on a worn glove: No surface is needed for this type, and typing can happen in “stealth mode”, e.g. while the hand is in the pant’s pocket. Tactile feedback is provided implicitly by finger contact. But at least with today’s technology a glove is essential to implementation.

**Method of key press detection:** What does the device actually register – the touch of a surface, an interruption of a light beam etc. This has a big impact on how robust the method is, i.e. whether actions that were not intended to be key presses might be recognized as such.

**Number of discrete keys:** How many keys does the method and/or device have? Is this number a hard limit or can it be increased easily? This is important if other or expanded alphabets like Chinese are to be accommodated. We denote a device that has similar characteristics as a regular

keyboard as having 52 keys and a plus “+” symbol stands for expandability. (The common typewriter has 52 keys while modern keyboards usually have 101 or more keys.)

**Key-to-symbol mapping DOF:** Does each key correspond to exactly one symbol/character (1 degree of freedom, DOF), or is it a one-to-many characters mapping (1.5 DOF, see section 4.2), disambiguated by either temporal methods (multiple successive keystrokes), by statistical prediction or by chording methods (multiple keys pressed simultaneously produce one character).

**Temporal significance interval for key press:** For the device to register a key press, how long does the key have to be pressed? On a physical keyboard, a key does not have to be depressed for a noticeable amount of time, but this might be different for virtual keyboards. Obviously, this has an immediate impact on the potential typing speed with the VK.

**Number of discrete operators:** Are all ten fingers used to operate the keys of the virtual keyboard or is it only the index finger, thumb etc. that can press a key? The number of discrete operators influences the parallelism that the human can utilize and therefore the potential bandwidth achievable with this device or method.

**Operator-to-key mapping:** Can any operator press any key or only a subset thereof? There are three different cases: A one-to-one mapping means there are as many keys as operators, and each operator works with the one key only. A one-to-many mapping is exemplified by how most people touch-type on a keyboard: One finger is responsible for a set of keys, and the key set for a finger does not intersect with any other finger's set, i.e. one key is always pressed by the same finger. Finally, a many-to-many mapping allows any operator to press any key. It must be noted that many devices theoretically have a many-to-many mapping, but individual users tend to operate a specific key with always the same operator. This is contrasted by the musical piano keyboard, where the keys are pressed by whatever operator is nearest by at any moment in time. (The authors are not aware of any interaction methods that use a many-to-one mapping.)

**Operator-key switch time:** This is related to the number of discrete operators that can be employed with the VK, but it is an independent quantity. It gives an idea about the human factors aspect of the time between pressing two different keys. A physical keyboard has a very small switch time, especially when two different fingers are used: We can hit two keys almost simultaneously. On a thumb-operated cell phone pad on the other hand, it might take a more significant amount of time to switch from one key to the next. Fitts Law [9] provides a widely accepted method for quantitative analysis. Due to the complexity of this quantity (it varies for different operators, keys, and operator-key sequences) we give only qualitative guidelines.

**Feedback:** Does the VK sport a feedback mechanism other than characters appearing on a screen? If so, what human sensors are used? There might be a time delay between the typing action and the character and/or feedback being visible (audible...) by the user – how big is this delay? The usability and “feel” of the VK will strongly depend on good feedback characteristics.

**Visual incarnation of a keyboard:** Is a projection or display mechanism available or feasible? With what technology? Visualization of the “keys” is an important consideration for novice users.

**Familiarity:** Is the VK an entirely new method to input text, does it have a remote relation to conventional keyboards (same layout but does not support all its affordances), or is it very much in the style of a physical keyboard? This will have a big impact on how easily users can transition to this kind of virtual keyboard, and in our opinion also on the broad acceptance of the VK.

**Estimated bandwidth:** This can of course be only a rough guideline to how many characters a human will likely be able to input per time when using this kind of device or method. We state the bandwidth in characters per minute (cpm). To compare to words per minute (wpm), a conversion factor of 5 can be used in accordance to the mean word length of 5 characters per word for the English language. We report raw key stroke numbers, not the number of characters after word completion or related methods.

**Invisibility:** How apparent is the device and its use to other people? Can the VK be used in “stealth” mode, without disturbing others or even without being noticed? The next three criteria are very important UI characteristics, yet limited information about almost all devices and methods rendered a comparison impossible.

**Cost:** What are mass production costs of the device? We put this in relation to the cost of a physical keyboard, which ranges in the tens of dollars.

**Reliability and robustness:** We are used to perfect accuracy of physical keyboards – no key press is registered unless we actually press a key, and no key press goes unnoticed. VKs might have non-zero false positive and false negative rates, for example caused by non-human disturbances to the recognition process or by borderline key press actions by the human.

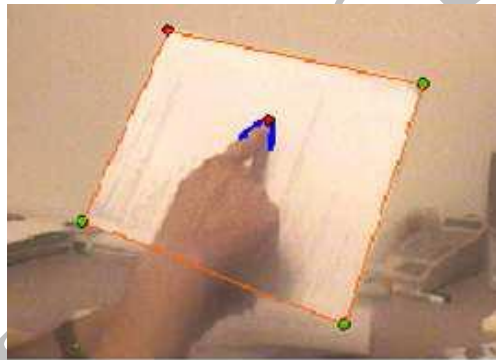
**Accuracy:** Strongly related to this is the accuracy with which the intended key press is recognized. For this measure, we regard every key as having a unique mapping to a character – a one-to-many key mapping has an additional effect on the final, perceived accuracy.

# VIRTUAL KEYBOARDS: METHODS AND DEVICES

The following subsections explain the main characteristics of each VK2. Some criteria from section 5 are addressed exclusively in the tabular comparison in subsection 6.13.

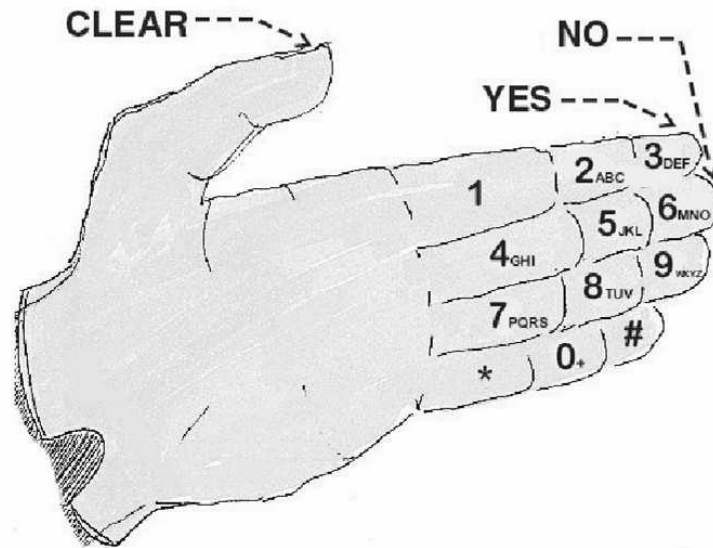
## Visual Panel

The Visual Panel [32] consists of a camera and a sheet of paper. The location of the extended index finger in reference to the paper is located with computer vision means. The primary application is a mouse pointer, clicking is achieved by resting the fingertip in its current position for three seconds. The authors demonstrated text entry by interpreting pointer locations as the keys of a keyboard, which were printed on the sheet of paper. An audible notification signals the recognition of a character after the 3 second significance interval.



## Finger-Joint Gesture Wearable Keypad

The FJG [11] suggests viewing the phalanges of the fingers (besides the thumb) of one hand as the keys on phone keypad. The thumb is used to press the virtual buttons. This is similar to Thumbcode, but it solely relies on word disambiguation to produce more than 12 characters. Yet the drawback of this 1.5 DOF key-to-symbol mapping might be mitigated by the familiar layout. Also, less complex hand configurations might be less tiring for the user. Just as Thumbcode, FJG has no user feedback method beyond skin contact sensations.



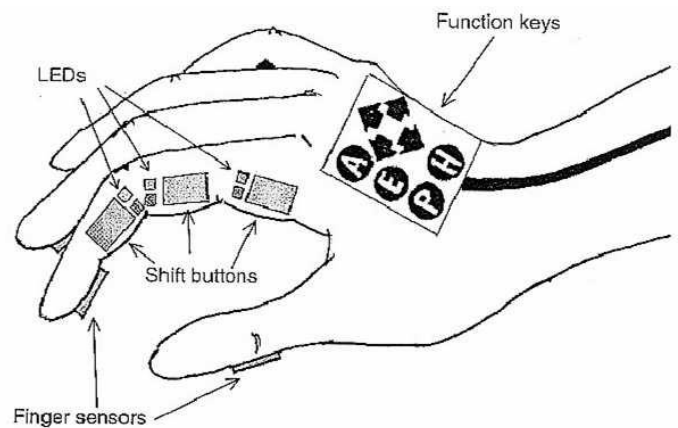
## Thumbcode

The “Thumbcode” method described in [21] defines the touch of the thumb onto the other fingers’ phalanges of the same hand as key strokes. Consequently there are 12 discrete keys (three for each index, middle, ring finger and pinky). To produce up to 96 different symbols, the role between keys and operators is broken up: The four fingers can touch each other in eight different ways, each basically representing a mode, or modifier key that affects the mapping for the thumb touch. Tactile user feedback is implicit when touching another finger with the thumb. A glove implementation was tested by the author.



## Chording Glove

The Chording Glove [22] employs pressure sensors for each finger of the right hand in a glove to implement a chording input device. Almost all possible finger combinations are mapped to symbols, making it potentially hard to type them. Additional “mode switches”, located along the index finger, are used to produce more than the 25 distinct characters. Yet user experiments suggest otherwise: rates of up to 19 wpm are achieved after ten training sessions “with no signs of leveling off”.



## FingeRing

FingeRing [10] uses accelerometers on each finger to detect surface impacts. In the wireless version depicted in the figure below these rings communicate with a wrist-mounted data processing unit. The interaction method is designed for one handed use, but could be extended to two hands with obvious implications. In the current version, the finger movements to produce one character are extensive: two chording patterns have to be typed within a time interval, each consisting of a combination of fingers hitting the surface. Due to this piano-style typing method, users with prior piano experience fare much better with this device; in fact, the full 2-stroke chord mapping is rendered too difficult for novice users.



## TouchStream

The TouchStream keyboard stretches our definition of a VK as it has keys printed on the surface. Yet the underlying technology permits software configuration of the sensed areas, equal to the multi-point touchpad described in subsection 6.6. Despite conventional touch-typing the TouchStream affords a number of chording patterns as alternatives to modifier keys. These patterns are pressed by one hand (anywhere on the pad) while the other touches the key that is to be modified.





## Multi-Point Touchpad

DSI Datotech Systems offers one of the few touchpads that reports up to ten surface contacts and their pressure forces independently and simultaneously [5]. While it has not been implemented yet, one could use the 20x15cm large device to transfer the traditional keyboard modality in a one-to-one fashion to an interactive, software-configurable surface. Inherent to this device are the same user feedback methods as for any of the devices employing tabletop units: finger surface impacts.



The multi-point touchpad, see subsection 6.7.

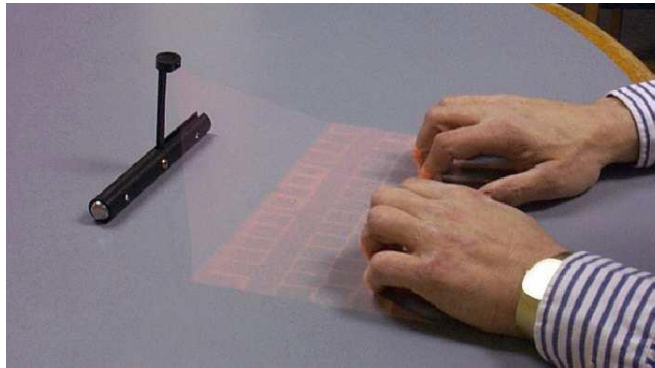
## VType

VType [7] detects the key stroke of each finger “in the air” with a data glove (fiberoptical curvature detection). Different locations of the key strokes are not distinguished, only which finger pressed a key. Instead, disambiguation with standard statistical methods on the word and sentence level solves the 1.5 DOF mapping problem. There is currently no feedback mechanism incorporated into the VType prototype.



## VKey

Virtual Devices Inc. recently announced a combined projection and recognition VK [29]. Little is known about this device, but their press release suggests that visual sensors (cameras) detect the movement of all ten fingers. Just as the VKB device, the VKey also consists of a tabletop unit and feedback is the tactile sensation of hitting a surface.



## VKB Projection

The virtual keyboard technology developed by VKB [30] is a tabletop unit that projects a laser image of a keyboard on any flat surface. Infrared cameras detect key strokes of all ten fingers. Word disambiguation techniques are employed despite this 1 DOF mapping. Therefore, our guess is that engagement of all distinct key locations is detected, yet with a fairly low accuracy. These two characteristics in combination should result in fairly good recognition rates. Surface impact of the fingers serves as typing feedback.



## Scurry

Tiny gyroscopes on each finger are the sensing technology in Samsung's Scurry [23]. The prototype suggests that these finger rings communicate with a wrist-mounted unit where the data is processed. Not much is known about this device, yet our guess is that finger accelerations and relative positions are detected, making it possible to distinguish multiple key targets per finger. We further guess that a surface impact is required to register a key stroke, also making for the primary sensory feedback to the user. Little LEDs on the rings potentially provide additional feedback.



## Senseboard

The Senseboard [24] consists of two rubber pads that slip onto the user's hands. Muscle movements in the palm are sensed (with unspecified, non-invasive means) and translated into key strokes with pattern recognition methods. All further information (obtained from the company's web site) can be found in the tabular comparison. The only feedback other than characters appearing on a screen comes from the tactile sensation of hitting the typing surface with the finger.



## Tabular Comparison

	Method or device	Number of discrete keys	Number of discrete operators	Key-to-symbol mapping DOF	Operator-key-mapping. *: potentially n-n	Operator-key switch time	Familiarity	Temporal significance interval for key press	Estimated bandwidth in cpm	Invisibility (+ high, o medium)	Visual incarnation of keyboard
Visual Panel	d	52+	1	1	1-n	Medium	Medium	3s	20	o	fixed
Finger-Joint Gesture	m	12	1	1.5	1-n	High	Medium	0	100	+	no
Thumbcode	d/m	12/96	1+4	1.5	1-n	High	Low	0	70	+	no
Chording Glove	d/m	5	5	1.5	1-1	Medium	Low	0	80	o	no
FingeRing	d/m	5	5	1.5	1-1	120ms	Low	120ms	120	+	no
Touchpad	d	52	10	1	n-n	Low	High	0	250	o	fixed possible
TouchStream	d	52	10	1	1-1	Low	High	0	250	o	yes
VType	d/m	10	10	1.5	1-1	Low	Medium	0	100	o	with HMD
VKey	d	52+	10	1	1-n*	Low	High	0	250	o	yes
VKB Projection	d	52+	10	1	1-n*	Low	High	0	250	o	yes
Scurry	d	52	10	1	1-n*	Low	High	0	250	o	no
Senseboard	d	52+	10	1	1-n*	Low	High	0	250	+	no

## CONCLUSIONS

We first gave an overview of the range of input devices and methods for alphanumeric data. We then had a closer look at touch-typing as input method and highlighted its benefits. This was followed by a survey of the state of the art of touch-typing interfaces, or virtual keyboards. We found that the trend goes towards retaining the original keyboard metaphor as closely as possible. Our conclusions are that while the keyboard is often regarded as an antique method that is unsuitable to modern computing devices, a number of characteristics are inherent in the way we use it that make it preferable over alternative methods. Input with keyboards is and will be an important user interface modality for computers for decades to come.

www.vtuics.com

## REFERENCES

- [1] Alden, D.G., Daniels, R. W., Kanarick, A.F. Keyboard Design and Operation: A Review of the Major Issues. *The Journal of the Human Factors Society*, vol. 14, 4, pp. 275- 293, 1972.
- [2] Blinkenstorfer, C. H. Graffiti. *Pen Computing*, 30–31, Jan. 1995.
- [3] Daniel, K. W. MurphyWas a Captioner.  
<http://www.robson.org/gary/captioning/murphy.html>
- [4] Davis, J. R. Let Your Fingers do the Spelling: Implicit disambiguation of words spelled with the telephone keypad. *Avios Journal*, 9, 57-66, March 1991.
- [5] DSI Datotech Systems Inc.: Multi-Point Touchpad; <http://www.dato.com>
- [6] Dvorak, A. There is a Better Typewriter Keyboard. *National Business Education Quarterly*, vol. 11, pp. 51-58, 1943.
- [7] Evans, F. and Skiena, S. and Varshney, A.; VType: Entering Text in a Virtual World, submitted to *International Journal of Human-Computer Studies*
- [8] FingerWorks, Inc. TouchStream Stealth. <http://www.fingerworks.com>
- [9] Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. In *Journal of Experimental Psychology*, vol. 47, pp. 381-391. 1954.
- [10] Fukumoto, M. and Tonomura, Y.: "Body Coupled FingeRing": Wireless Wearable Keyboard; *ACM CHI 1997*.
- [11] Goldstein, M. and Chincholle, D.; Finger-Joint Gesture Wearable Keypad; in *Second Workshop on Human Computer Interaction with Mobile Devices*, 1999.
- [12] Klockenberg et al. *Rationalisierung der Schreibmaschine und ihrer Bedienung*. Springer Verlag, Berlin, 1926.
- [13] Liebowitz, S. J. and Margolis, S. E. The Fable of the Keys. *Journal of Law & Economics* vol. 33, April 1990.
- [14] MacKenzie, I. S., and Soukoreff, R. W. A model of twothumb text entry. *Proceedings of Graphics Interface 2002*, pp. 117-124.
- [15] MacKenzie, I.S., Soukoreff, R.W. Text Entry for Mobile Computing: Models and Methods, Theory and Practice. *Human-Computer Interaction*, 2002. To appear.  
<http://www.yorku.ca/mack/hci3.html>

- [16] Mankoff, J. and Abowd, G. D. A word-level unistroke keyboard for pen input. ACM Symposium on User Interface Software and Technology 1998.
- [17] Norman, D. A., Fisher, D. Why Alphabetic Keyboards Are Not Easy to Use: Keyboard Layout Doesn't Much Matter. The Journal of the Human Factors Society, vol. 25, 5, pp. 509-519, 1982.
- [18] Noyes, J. The QWERTY Keyboard: A Review. International Journal of Man-Machine Studies, 18, 265-281, 1983.
- [19] Perlin, K. Quikwriting: Continuous Stylus-based Text Entry. ACM Symposium on User Interface Software and Technology 1998.
- [20] Plant, S. On The Mobile. Motorola Report, 2001.  
<http://www.motorola.com/mot/documents/0,1028,297,00.doc>
- [21] Pratt, V. R., Thumbcode: A Device-Independent Digital Sign Language.  
<http://boole.stanford.edu/thumbcode>
- [22] Rosenberg, R.: Computing without Mice and Keyboards: Text and Graphic Input Devices for Mobile Computing; PhD Thesis, Dept. of Computer Science, University College, London, 1998.
- [23] Samsung Scurry; PCWorld article at  
<http://www.pcworld.com/news/article/0,aid,70568,00.asp>
- [24] Senseboard, <http://www.senseboard.com>
- [25] Shannon, C. E. Prediction and Entropy of Printed English. Bell System Technical Journal, 30, 51-64.
- [26] Shneiderman, B. The Limits of Speech Recognition. Commun. ACM, 43, 9, 63-65, Sept 2000.
- [27] Silfverberg, M., MacKenzie, I. S., and Korhonen, P. Predicting text entry speeds on mobile phones. Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 9-16, 2000.
- [28] Tegic Communications. T9 Text Entry. <http://www.t9.com>
- [29] Virtual Devices, Inc., VKey; [http://www.virtualdevices.net/press\\_release.htm](http://www.virtualdevices.net/press_release.htm)
- [30] VKB Ltd., <http://vkb.co.il>
- [31] Zhai, S., Hunter, M. and Smith, B.A. The Metropolis Keyboard – An Exploration of Quantitative Techniques for Virtual Keyboard Design. UIST 2000.

- [32] Zhang, Z., Wu, Y., Shan, Y., and Shafer, S. Visual Panel: Virtual Mouse, Keyboard and 3D Controller with an Ordinary Piece of Paper. ACM Workshop on Perceptive User Interfaces, Nov. 2001.

[www.vtuics.com](http://www.vtuics.com)