Univ Access Inf Soc (2017) 16:51–72
DOI 10.1007/s10209-015-0433-0

LONG PAPER

Text input for motor-impaired people

Ondrej Polacek[1] · Adam J. Sporka[1] · Pavel Slavik[1]

Published online: 21 December 2015

Abstract This paper provides an overview of 150 publications regarding text input for motor-impaired people and describes current state of the art. It focuses on common techniques of text entry including selection of keys, approaches to character layouts, use of language models, and interaction modalities. These aspects of text entry methods are further analyzed, and examples are given. The paper also focuses on an overview of reported evaluations by describing experiments, which can be conducted, to assess the performance of a text entry method. Following this overview, a summary of 61 text entry methods for motor-impaired people found in the related literature is presented, classifying those methods according to the aforementioned aspects and reported evaluation. This overview was assembled with the aim to provide a starting point to the new researchers in the field of accessible text entry. The text entry methods are also categorized according to the suitability for various conditions of the users.

Keywords Text input · Text entry methods · Motor-impaired people · Overview · Summary

1 Introduction

Text input is a common activity of many ICT devices such as laptops, phones, or tablets. Even though most people find entering text easy and natural, it is challenging for people with certain disabilities. In order to support inclusion of people with disabilities into society, it is important to offer text entry methods which are appropriate for their needs. This paper gives an overview of existing methods and techniques covering a broad range of groups of motor-disabled people.

Many different physical impairments exist, causing various disabilities from deteriorated finger dexterity to complete paralysis, resulting in ample range of obstacles to use the devices intended for majority population.

Physical impairments are consequences of traumatic injuries, diseases, and congenital conditions. Spinal cord injuries can cause malfunction of legs (paraplegia) or all limbs (quadriplegia). Loss or damage to upper limbs is another traumatic injury that affects work with computers. People with cerebral palsy can experience spasms, involuntary movement, impaired speech, and even paralysis. Muscular dystrophy is a disease in which the muscles are progressively degenerated and can also lead to paralysis. People with multiple sclerosis experience different symptoms such as tremors, spasticity, or muscle stiffness. Spina bifida causes motor difficulties that can lead to paralysis. Amyotrophic lateral sclerosis causes slowness in either movement or speech. Elderly people can be often inconvenienced by arthritis. Pain in joints affects fine motor control. Parkinson's disease and essential tremor cause uncontrollable tremors and affect the voice in more severe cases as well.

To provide an access to computing technology to such a diverse group of people, many techniques and interaction modalities are used, including those facilitating the text entry.

A wide range of text entry methods has appeared in recent years. One may not determine a dominant text entry method for motor-impaired people. Figure 1 shows

✉ Adam J. Sporka
sporkaa@fel.cvut.cz

[1] Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo nam. 13, 12135 Prague 2, Czech Republic

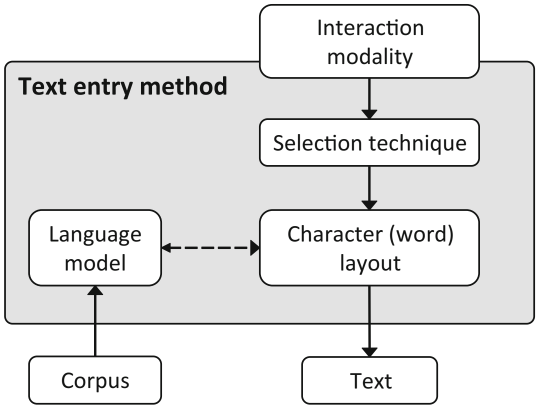


Fig. 1 A simple model of a text entry method

common features of a typical text entry method. The essential function of every text entry method is to provide means for inputting text by selecting individual characters or words. Common techniques for character and word selection are described in Sect. 2. A character or word is selected from a method-specific layout or distribution which is discussed in Sect. 3. The layout of characters or words usually depends on a language model. The use of language models is described in Sect. 4. In order to make the character selection possible, an appropriate interaction modality has to be used. Interaction modalities suitable for motor-impaired people are summarized in Sect. 5.

Text entry methods are usually subject to experiments and evaluations. Possible experimental setups and typical evaluations are discussed in Sect. 6. In Sect. 7, 61 methods identified in related literature are summarized. The methods are classified according to the type rate, selection technique, character layout, language model, and interaction modality. Discussion regarding the experiment setups of these methods is also presented in this section.

1.1 Related overviews

Several overview papers exist focusing on text entry in augmentative and alternate communication. For example, Boissiere and Dours [12] published in 2003 an overview of existing writing assistance systems focusing mostly on prediction and language modeling. A short overview with a special focus on conversation modelings was published by Arnott [4]. Trewin and Arnott [131] describe mostly specialized keyboard hardware and on-screen keyboards for motor-impaired people. Other over-views overviews focus on ambiguous keyboards for augmentative and alternate communication [46], scanning systems [66, 96], and eye typing [74].

The overviews presented above describe only a subset of related literature focusing on a particular technique or modality. This paper, on the other hand, gives a summary of text input techniques and methods for motor-impaired people, describing and reviewing evaluations of text entry methods reported in the literature.

2 Selection techniques

On the first typewriters, typing one character was simply done by selecting one key. When the *Shift* key was added, the number of characters to enter became greater than the number of keys the typewriter provided. Mobile phones with 12 keys allowed entering almost the same range of characters as a standard PC keyboard with approximately 100 keys.

As mentioned above, various impairments with rather diverse consequences exist. While some people are completely excluded from using the PC keyboard, a number of people can still use it, albeit achieving slow type rates. Reducing the number of keys might be one way of making the text input accessible as it results in devices with smaller physical footprint on which a shorter (or zero) distances need to be traveled by users' hands and thereby increasing the type rate (or making the text entry even possible, such as the single-switch devices).

Every text entry method has some kind of selection technique to determine the character to be entered. In their survey of related literature, the authors have identified the main selection techniques as follows—*direct selection*, *scanning*, *pointing*, and *gestures*.

2.1 Direct selection

Direct selection is a technique, which enables the user to select directly a key out of a set of keys. The set of keys available for motor-impaired people is usually limited. When reducing the set of keys, three basic techniques can be used—chording keyboard, ambiguous keyboards, and encoding.

Chording keyboards reduce the number of keys; however, they require the ability to press multiple keys at the same time. Different combinations of pressed keys correspond to different characters typed, which results in relatively high type rates [61]. This selection technique, however, is rarely used for motor-impaired people as reduced dexterity of their hands hinders them from accurately pressing multiple keys at the same time. A number of chording keyboards exist with different key-to-character mappings. An example of a possible mapping is depicted in Fig. 2. Letters "a" and "b" are entered by single keys (K1 and K2, respectively) though the letter "c" is entered by pressing K1 and K2 simultaneously.

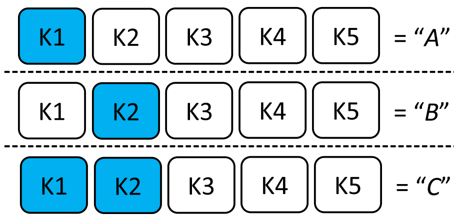


Fig. 2 Chording keyboard with five keys. The *dark* (*blue*) background indicates the pressed keys

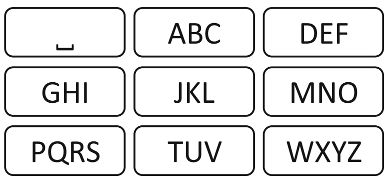


Fig. 3 Ambiguous keyboard. Multiple letters are assigned to one key

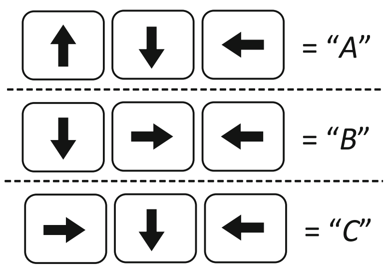


Fig. 4 An example of encoding. *Four arrow* keys (*left, right, up, down*) are used to encode letters in this example

Ambiguous keyboards [65], such as T9 [32] or Multitap [72, 99], are very popular among motor-impaired people. In these keyboards, the alphabet is divided into several groups of characters and each group is assigned to one key (see Fig. 3). Ambiguous keyboards are described in detail in Sect. 4.3.

Encoding [12, 45] is another technique which aims at reducing the number of keys. Each character corresponds to a unique sequence of key presses. The sequence is usually referred to as "code." Examples for this case include binary spelling interfaces using Morse code [127] and Huffman code [130], where only two keys are used. Other examples are Minimal Device Independent Text Input Method (MDITIM) [42] or "Up–Down–Left–Right" (UDRL) [18], both using four direction keys. An example of encoding is shown in Fig. 4 where letters are assigned to a unique sequence of four arrow keys.

2.2 Scanning

When only a very low number (one or two) of keys are available, *scanning* can be used in a text entry method. Scanning systems and keyboards have been studied extensively in the past decades. Scanning refers to an item selection technique in which a number of items are highlighted sequentially until the desired item is selected, and the corresponding command is executed (e.g., a letter is typed). Scanning is based on two atomic operations: *scan step* and *scan selection*. The step operation highlights the next item in a predefined order, while the selection operation executes a command assigned to the highlighted item.

Scanning text entry methods can be categorized according to two aspects—modes and techniques. A *scanning mode* determines mapping of user input on scan step and selections (e.g., [84]). A *scanning technique* describes how items are grouped and how the scanning proceeds among the items.

2.2.1 Scanning modes

In the simplest case, scanning requires only one unique signal from the user (further referred to as switch), which is mapped either to step or to selection. The other operation is then triggered after a predefined scanning interval is reached. Based on the mapping, several scanning modes can be distinguished. The most prevalent are as follows:

Automatic scanning. The automatic scanning is the most common mode. The selection is controlled by the user input (switch activation), and the step is triggered automatically after a scanning interval expires. An example is shown in Fig. 5.

Step scanning. Step scanning [84] (see Fig. 6) is similar to the automatic scanning, and the control of selections and steps is reversed: Steps are controlled by the user, and selections are automatic.

Self-paced scanning. Self-paced scanning [24] (see Fig. 7) distinguishes between single and double switch activations. Double switch activations correspond to two consecutive switch activations issued within a short timeout. Then, the single switch activation is used as a scan step and double switch activation as a scan selection.

Inverse scanning. Inverse scanning [84, 85] (see Fig. 8) requires a switch with two states (e.g., button is pressed or

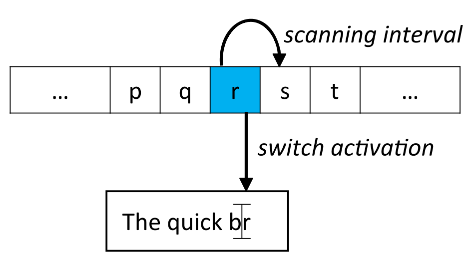


Fig. 5 Automatic scanning. Selection is made by switch activation and step by a scanning interval

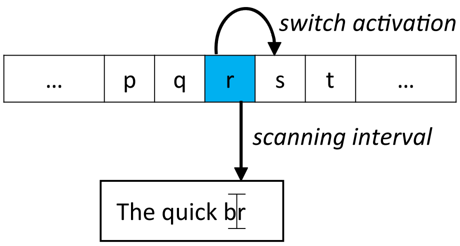


Fig. 6 Step scanning. Selection is made by a scanning interval and step by switch activation

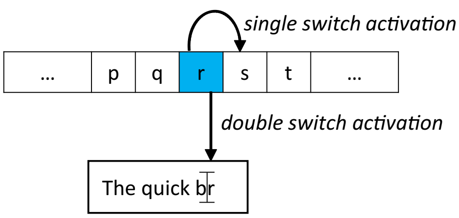


Fig. 7 Self-paced scanning. Selection is made by a double switch activation and step by single switch activation

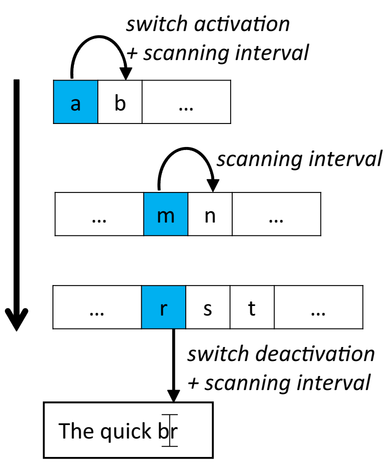


Fig. 8 Inverse scanning. Scanning starts after switch activation (e.g., pressing a button switch). Item is selected when the switch is deactivated (e.g., releasing a button switch)

released). When the switch is activated (e.g., a button is pressed), automatic scanning is started. Once deactivated (e.g., a button is released), the selection is made by waiting for the scanning interval.

2.2.2 *Scanning techniques*

Several scanning techniques are used not only for text entry but also for menu selections or browsing the contents of a menu.

Linear scanning (e.g., [138]) is probably the simplest technique. Items are sequentially highlighted in one group until the desired item is selected. An example of typing the

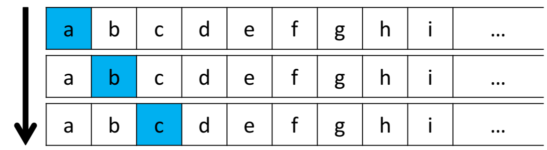


Fig. 9 Linear scanning. Letters are highlighted sequentially in each scanning step

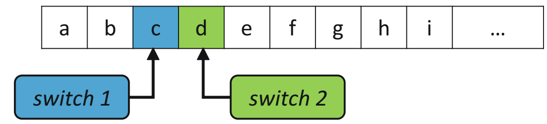


Fig. 10 Two switch scanning. Two letters are highlighted in one scan step

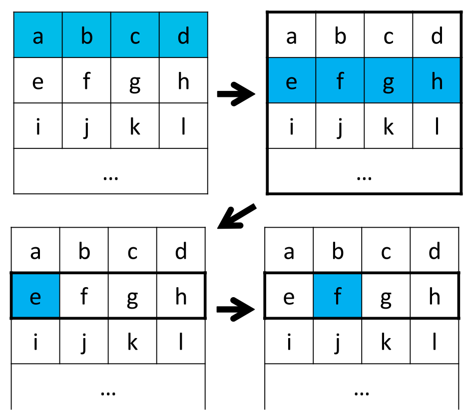


Fig. 11 Row-column scanning. An example of selecting the letter “f” by one row and one column step

letter “c” is depicted in Fig. 9. When two switches are available, the scanning interval is usually replaced by the second switch. Another approach, which uses two switches, is to keep the scanning interval and offer the user the possibility to select the current highlighted nth item or the next $(n + 1)$th item [107]. In example depicted in Fig. 10, either “c” or “d” can be entered within one scanning interval by switch 1 or switch 2, respectively.

Linear scanning is very slow, especially for items at the end of the sequence. To address this issue, *row-column scanning* (e.g., [49, 112]) can be employed. In this technique, the items are organized in a matrix and selecting an item is done in two levels. In the first level, rows are sequentially highlighted until the selection is made, and then, items in the selected row are linearly scanned. The process of selecting letter “f” is depicted in Fig. 11.

Three-dimensional scanning (or *group scanning*) [22, 59] reduces the number of scanning steps by adding one more level. In this level, groups of characters (or quadrants, see Fig. 12) are sequentially highlighted until the selection is made. Then, standard row-column scanning is employed.

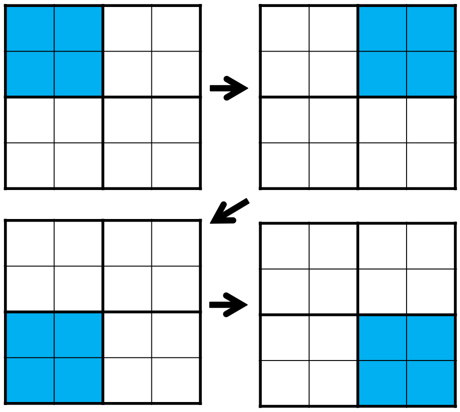


Fig. 12 Three-dimensional scanning

Binary scanning (or *dual scanning*) [24, 37] recursively splits items into two halves, prompting the user to select one of the two options, until a single item is highlighted and selected. This technique is similar to binary search well-known from basic programming algorithms. *N*-ary scanning [103] is the generalization of the binary scanning where the range of choices is recursively split into *N* groups.

Ternary scanning was found to be optimal among other *N*-ary scanning techniques for the use case of character input. An example of entering the letter "n" by ternary scanning is depicted in Fig. 13. In the first level, the alphabet is split into three groups "a–h," "i–q," and "r–z." The scanning proceeds to the second group where the selection is made. In the second level, "n" is selected within the second scan step.

Containment hierarchical scanning [6] is a model that describes any scanning system in terms of an acyclic graph. In such graph, hierarchical relations are defined—groups containing subgroups and single items. The graph defines multiple levels of scanning starting in root and proceeding down to leaves which contain characters. As soon as the leaf is reached, the corresponding character is entered. For example, the binary Huffman tree [40] was used in work by Baljko and Tam [6] and Roark et al. [108]. An example of scanning such tree spatially organized into matrix is depicted in Fig. 14.

The selection of scanning technique depends on the input modality, user abilities, and number of items that are scanned through. Input modality and user abilities influence the time needed to activate the switch and thus make the selection. The scanning interval has to be longer than the time required for selection. Scanning steps and scanning selections should be balanced in order to maximize the type rate. Scanning selections should be kept low as each switch activation requires the user to perform an action. Too frequent switch activation might be uncomfortable for the user.

The number of items also influences the selection of scanning technique. For low number of items, linear scanning is good enough (e.g., scanning ambiguous keyboards [67]). For an alphabet comprised of basic letters and characters, row-column scanning is usually used. When a higher number of items are required, the three-dimensional scanning is a good option [22].

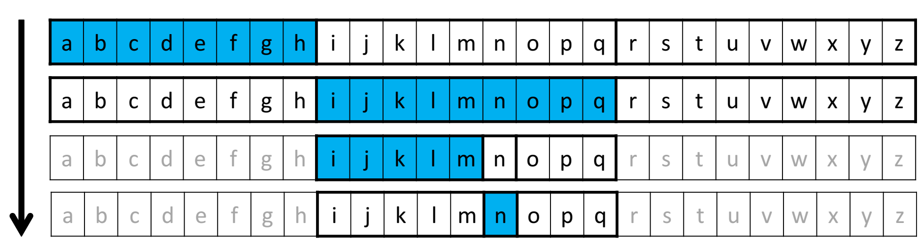
2.3 Pointing and gestures

In some text entry methods, a pointing device controlling a pointer is used to enter the text. While many motor-impaired people cannot control directly the mouse, they still can emulate it by trackballs, joysticks, head tracking, gaze, etc. Some of these modalities, however, do not support selection (e.g., head tracking, gaze interaction). Three common solutions to this problem exist—*dwell time*, *multimodal interaction*, and *gestural input*.

In the *dwell-time method*, a selection is generated after a predefined time-out when the pointer is kept within a small radius. The dwell-time method is capable of simulating one selection only (usually mapped to the left mouse click). Moreover, this method raises the so-called Midas touch problem [43] which relates to the fact that the selection is always triggered when the cursor is not moving. This may lead to many involuntary selections as the user cannot rest without making one. The Midas touch problem can be solved by either adding areas where the user can stop the cursor [10] or displaying a pop-up menu after the dwell time expires [102].

In the *multimodal interaction* approach, a different modality is used to generate the selection. Examples for this can be the tracking of head features [28, 132, 134], speech recognition [60, 110, 136], nonverbal vocal commands [102], teeth clicking [151], etc. Both dwell-time

Fig. 13 Ternary scanning



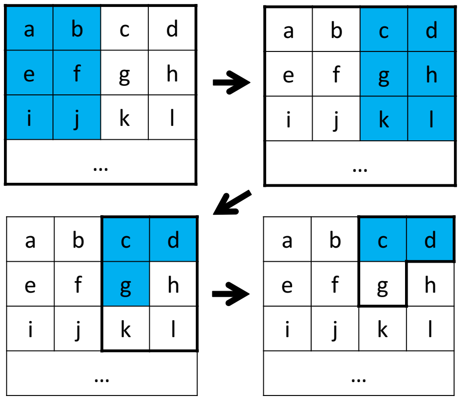


Fig. 14 Containment hierarchical scanning

method and multimodal interaction can be used as a universal selection method with any pointing device.

In *gestural input*, strokes with the pointing device are frequently transformed to text. Strokes can correspond to a character (e.g., EdgeWrite [144, 146, 147]), word (e.g., Quikwriting [100], SHARK [149], or Continuous EdgeWrite [79]), or even longer parts of the text (e.g., Dasher, [140]). The gestural input cannot be usually used as a universal selection method for a pointing device with an exception to gaze interaction [89].

The performance of pointing devices is modeled by Fitts' law [27] which can be used for predicting movement times based on distance to travel and width of a target. Fitts' law is not only a robust predictor for hand movement, but also for the pointing with mouse and joystick [64], head [1], or tilt [68]. On the other hand, when applied to gaze interaction, robustness declines due to saccadic eye movements. These movements make the movement time independent of distance [116]. In text entry systems, Fitts' law is often used as a heuristic for approaching the optimal layout of an on-screen keyboard operated by a pointing device, e.g., [2, 70, 82, 119, 150].

3 Character layouts

The ultimate goal of each text entry method is to maximize the type rate. In order to achieve this goal, researchers strive to find an optimal layout of letters or optimal code for given text entry technique, interaction style, or interaction method. A well-known example is the Dvorak simplified keyboard [17], which is claimed to minimize finger motion in order to increase the type rate of a typewriter.

Optimal layouts are usually built by analyzing frequencies of words and letters in a given corpus based on a language. Thus, a common property of optimal layouts is their language dependency. Alphabetical layouts seem to be language independent and suitable for novice users. However, the language independence of alphabetical layouts is questionable as many languages contain special characters or diacritical marks. No consensus exists regarding the suitability for novice users. A study by Norman and Fisher [92] found no improvement when comparing alphabetical to a randomized layout. Other studies [30, 118], on the other hand, confirmed that the alphabetical layout results in better performance of novice users.

The character layouts can be divided into two categories according to character distribution—static and dynamic [12]. By the term character distribution, not only visual representation of the text entry method, but also sequence of operations, which has to be entered to type a character, are denoted. These may refer to a layout of an on-screen keyboard as well as to characters encoded into a sequence of keystrokes. In static distributions, this sequence of operations remains the same while typing. In dynamic distributions, however, the sequence is gradually updated according to currently written context.

The difference between static and dynamic distributions can be demonstrated on two methods for mobile text input on a nine-button keypad. Both methods are controlled similarly as MultiTap method (e.g., [72, 99]) commonly available on mobile phones, in which each key is assigned to three or four characters and a character is entered by repeatedly pressing the same key until the desired character appears.

The Less-Tap [99] method uses a static distribution, where the sequences of letters on the keys are sorted according to their probability in English. LetterWise [72] uses dynamic distribution. In this case, sequences of letters on each key are sorted according to their current probability based on already written context. The probability is updated after a letter is entered. An example of typing the word "sky" is shown in Fig. 15. While the presentation of the layout to the user is static (due to the hardware of older mobile telephones), the distribution itself is dynamic.

A trade-off between static and dynamic distributions exists: The static distributions require less cognitive effort from the users as their behavior does not change with the context. The dynamic distributions require less physical effort as they minimize the number of necessary steps users need to perform at the cost of adapting the layout according to context. It can be seen as a contradiction of Nielsen's heuristics "Consistency and standards" and "Flexibility and efficiency of use."[1]

[1] http://www.nngroup.com/articles/ten-usability-heuristics/.

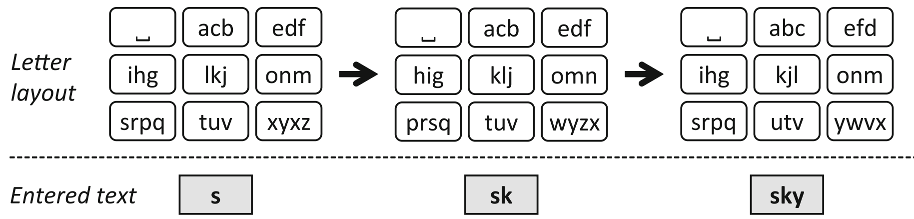


Fig. 15 Typing "sky" on a dynamic layout (the original method was intended for physical keyboards of the mobile telephones, and therefore, shifting of the letters on individual keys was not represented graphically)

3.1 Static distribution

Static distributions are not as cognitively demanding as dynamic distributions as the users may memorize the character distribution relatively easily. Some of the methods can be even eyes-free for an expert user, depending on the interaction modality used.

A static distribution is designed prior to using the text entry method, usually by analyzing static frequencies of single characters or *n-grams* (i.e., *n* successive letters). The distribution can either be a result of designer's creativity or computed by an algorithm. Finding the distribution is an NP-complete optimization problem [55], and thus, either a search heuristic or search space constraints need to be used. A common search heuristic is a genetic algorithm used for layouts of ambiguous keyboards (see Sect. 4.3), e.g., [30, 38, 47]. Fitts' Law can be used as a cost estimator for keyboards operated by pointing. Yin and Su [148] used a particle swarm optimization algorithm for the same purpose. Search space constraints were used in ambiguous keyboard design [30, 67] by forcing the layout to follow the alphabetical order.

3.2 Dynamic distributions

In dynamic distributions, the sequence of user inputs to enter a character is changed gradually according to the actual probability in current context [12]. The distribution is usually computed from a letter-level language model (see Sect. 4.2 for more details). This approach minimizes the number of user inputs needed to enter a character at the expense of higher cognitive demands on the user.

Dynamic distributions are mostly used in scanning systems by rearranging letters. This can be done either in the whole layout (e.g., [39, 56, 138]) or only locally. The local approach tries to retain letters close to their original positions by rearranging only parts of the layout similarly to aforementioned LetterWise [72] method (e.g., [83, 84, 88]). Another approach is to retain letter position static while making the scanning sequence dynamic [103]. Dynamic layouts may offer not only single letters, but also several highly probable continuations [101].

Text entry methods often combine dynamic and static aspects in order to lower the cognitive demands. An example of such combination is Dasher [140]. Although the letters are always displayed alphabetically in this method, their size changes dramatically according to their probability. Another dynamic keyboard, GazeTalk [44], shows dynamically the six most probable letters. When no letter is desired, a static full keyboard is available. SpreadKey [82] is a virtual QWERTY where low-probability characters close to currently typed characters are replaced by highly probable characters. pEYEwrite [135] is a method based on a hierarchy of three pie menus. While the first two menus are static, the last dynamically provides five most probable next letters.

4 Use of language models

Many different language models are used in a wide range of scientific disciplines. Thus, no exact definition of a language model exists. Generally, one may argue that the language model is a means for describing languages in a structured and consistent way. In this section, the use of language models in text entry methods is described. Two main techniques were identified, which are inevitably dependent on a language model—prediction (see Sect. 4.2) and ambiguous keyboards (see Sect. 4.3). They are not mutually exclusive and can be combined in a single text entry method.

Almost every text entry method inherently uses a language model—from simple static probabilities of characters to more sophisticated language models. While number of language models exists, three essential approaches can be found in the literature on text input [12, 29]—*syntactic, semantic, and statistical*.

Syntactic and semantic approaches store rules either in probability tables or as a grammar. The difference between these two approaches lies in the categorization of words (syntactic or semantic categorization). The semantic approach was employed, for example, in the work by Demasco and McCoy [16].

The *statistical* approach stores frequencies of *n-grams*. An *n*-gram is a sequence of *n* items. The item can be either a letter or a word. Based on the item type, one can then distinguish between letter-level *n*-grams [140] and word-level *n*-grams (e.g., Google *n*-gram corpus [31]). *n*-grams with length equal to one, two, and three are called unigrams, bigrams, and trigrams, respectively. The *order of the model* further refers to the longest *n*-gram contained in the language model. The probability of the next items is extracted from the model based on already written $n-1$ items. These written items (letters or words) are usually referred to as context.

The statistical approach to language modeling is the simplest one [29]. It prevailed in text entry methods published in the past decade, while the other two approaches diminished. It may be argued that it was because of the simplicity of the corpus acquisition of the statistical approach.

4.1 Statistical language models

Text entry methods usually use two types of statistical language models—letter level and word level. Obviously, the type of the language model depends on the type of *n*-grams stored (letters or words).

Letter-level language models are usually stored as a prefix tree, often called *Trie* (from re**trie**val). This data structure enables fast basic operations for *n*-grams—indexing, adding, and deleting. An example of a Trie is shown in Fig. 16. The 1st-order model stores only unigrams. It can be used for statically distributed layouts (see Sect. 3.1) as only static probability of letters is stored. In order to employ dynamic layouts and prediction, a higher-order model has to be involved. Several orders of the models are reported in the literature, e.g., 2nd-order [135], 3rd-order [13], 4th-order [72, 83], 6th-order [108, 109], or 8th-order [97]. With increasing order of the model, the performance of the prediction improves. However, the performance does not improve significantly after the 6th-order model [104], following the pattern consistent with the law of diminishing returns.

An interesting system based on letter-level statistical approach is described by Shieber and Baker [115]. In this system, words are completed from abbreviations—the user omits all vowels and consecutive duplicate consonants while typing. This approach saves about 30 % of keystrokes.

The basic model as described above works well for previously seen *n*-grams stored in the model. However, a text entry method often has to predict probabilities of all characters in a given context including *n*-grams that have not yet been stored in the model. This is the so-called zero-frequency problem [142] which can be solved using one of

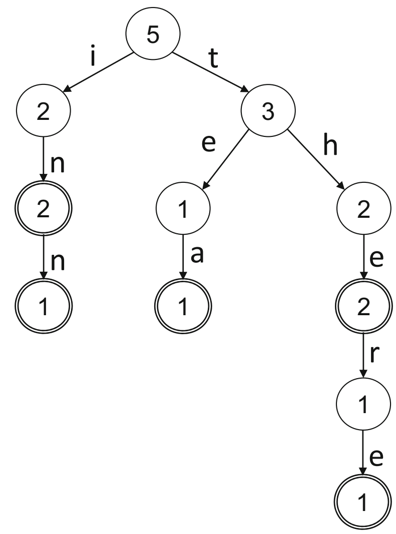


Fig. 16 An example of a letter-level language model in form of a prefix tree (often referred to as *Trie*). The model contains five words: "tea," "in," "the," "inn," and "there." The numbers in the nodes express frequency of letters

zero-frequency estimation predictor from adaptive data compression techniques. For example, a predictive text entry method, Dasher [140], uses prediction by partial match (PPM) algorithm [129] to deal with the zero-frequency problem: If the string of characters, which is the current context, is not present in the PPM language model, the model is queried for a shorter string until an instance is found.

Word-level language models are usually stored in tables [29]. The order of the model is rather limited because of high storage complexity. For example, a 3-gram dataset from Google *n*-gram corpus [31] contains about 23×10^9 entries. Even though the *n*-grams can be limited to only the most probable, text entry methods usually use only unigrams or bigrams.

4.2 Prediction

A text entry method can be accelerated by prediction, when a list of completions is updated with each character entered. The aim of a predictive system is to reduce the number of keystrokes per character by offering shortcuts to the most probable characters or words. The ranks of the offered entries are determined in the language model.

Prediction has been heavily used in text input for motor-impaired people. One of the first prediction systems, used in the Reactive Keyboard [15], predicted possible words according to the context that had been already written. An adaptive dictionary-based language model was used. Predicted candidates could be selected by the mouse cursor. Expert users of a QWERTY keyboard would be slowed down; however, such prediction is useful for poor typist or people with limited movement of upper limbs.

Another example is the VITPI system [11] which offered unambiguous parts of words found in a dictionary. In Dasher [140], letter-level prediction was used to alter size of virtual keys. A combination of letter-level and word-level language model was used in GazeTalk [44], which predicted six most probable characters and six words according to the current context. Prediction of longer chunks of text was studied in a work by Polacek et al. [101].

4.3 Ambiguous keyboards

Ambiguous keyboards may be defined as keyboards where multiple characters are assigned to a single key. The basic idea of ambiguous keyboards is to divide the alphabet into several groups of letters and then to assign each group to one key. The ambiguity lies in the fact that a group of letters is assigned to one key only. In order to select the desired letter from the group, a disambiguation process has to be executed. Two disambiguation processes are commonly reported in the literature—letter-level and word-level disambiguation.

4.3.1 Letter-level disambiguation

The letter-level disambiguation is used in the popular mobile text entry method—MultiTap and its variations [72, 99]. In this method, the user first selects the desired key and then disambiguates the character by multiple keystrokes. The character is then entered after a time-out is reached or another key is pressed. From the character selection point of view, the technique corresponds to two successive selections: a direct selection followed by step scanning (see Sect. 2.2).

In text input for motor-impaired people, the two selections in the disambiguation technique can be replaced to lower the number of required input signals or to better fit to the interaction modality used. For example, in a work by Mirro-Borras et al. [83–87], the selection process is carried out as follows: First, automatic selection is employed to select the key, and then, inverse scanning is used while keeping a switch activated. Note that after replacing the selection process by two successive steps of automatic scanning, the letter-level disambiguation turns into row-column scanning (see Sect. 2.2 for more details). The letter-level language model is used in their work to shuffle characters on individual keys according to probability, similarly to LetterWise method [72].

4.3.2 Word-level disambiguation

Word-level disambiguation was first used in the commercial T9 system by Tegic Communications [32]. This method was widely used in mobile phones with a 12-button keypad. Instead of entering a particular letter, the user selects a corresponding key which contains the letter. After entering a sequence of keys, the list of most probable words is shown. The list is a result of a disambiguation process which selects possible words from a word-level language model. The model is often referred to as a dictionary. This is the strength as well as the weakness of the method. Thanks to the dictionary disambiguation process, the method is very efficient in terms of keystrokes per character (KSPC). According to MacKenzie [62], the theoretical KSPC value is very close to 1. This value, however, does not include nondictionary words. When the user wants to enter a new word which does not exist in the dictionary, another method has to be used, which significantly slows the typing and increases the KSPC rate. The efficiency of the method is heavily dependent on the quality and completeness of the dictionary [33].

Many ambiguous keyboards were designed for physically impaired people. Kushler [53] describes an ambiguous keyboard in which the alphabet was assigned to seven keys, while the eighth key was used as a space key that initiated the disambiguation process. Tanaka-Ishii et al. [128] published a similar system, in which only four physical keys were used. Besides disambiguation, the text entry method was capable of predicting words. Harbusch and Kühn [38] presented a similar method in which the whole alphabet was assigned to only three keys and one key was used for executing a special command in a menu.

The *scanning ambiguous keyboard* [67] is an ambiguous keyboard with word-level disambiguation in which the direct selection of a key is replaced by automatic scanning. Because of its efficiency with minimum input signals, scanning ambiguous keyboards became quite popular among text entry methods for motor-impaired people. For example, Kühn and Garbe [52] used a four-key scanning ambiguous keyboard. Harbusch and Kühn [37] showed that the scanning ambiguous keyboard outperforms other scanning text entry methods. Belatar and Poirier [9] used three keys and developed a virtual mobile keyboard. In Qanti [25], three keys are mapped to the alphabet.

5 Interaction modalities

The number of different input modalities can be used to help motor-impaired people interact with a computer. This section summarizes the most prevalent modalities for text input. Some text entry methods are designed generically to a certain extent—they can use multiple modalities which share a certain feature. An example is the MDITIM method [42] designed for any device which can yield four directional signals. On the other hand, some of them are tailored to a single modality only (e.g., [89, 139]).

5.1 Mechanical interfaces

People with residual dexterity in hands can still use a PC keyboard. However, they usually have problems with chording input (pressing two or more keys simultaneously). This problem is solved by the accessibility options of most operating systems (e.g., sticky keys in Windows). Several special devices have been developed to decrease motoric demands on the user, for example, Maltron ergonomic keyboards.[2] Chubon and Hester [14] presented an optimized PC keyboard for typing with one finger only by rearranging letters. Felzer et al. [26] described a keypad with 4×5 layout which used an on-screen keyboard to enable typing.

Scanning is operated by a switch as already mentioned in Sect. 2.2. A number of different switches exist, such as mechanical switches (or buttons), pneumatic switches controlled by breath, and bite switches. An exhaustive list of switch types is given in a work by Ntoa et al. [96]. Some nonstandard modalities have been used as switches, for example, eye blinks [5, 113], eye movements [74], intentional muscle contractions [25], or nonverbal vocal interaction [103].

Game controllers were found to be ergonomic not only for gamers, but also for people with motor impairments. For example, the MDITIM method [42] can be controlled with a joystick. Dual joysticks, which are present on some game controllers, can be used to improve text entry rate [50, 141]. Joystick text entry with prediction for motor-impaired people was studied in a work by Song [120]. Felzer and Rinderknecht [23] used an eight-way direction pad on a game controller to enter a text. Another use of a game controller is reported in H4-Writer [73] and in Continuous EdgeWrite [79].

For motor-impaired people, the mouse is often replaced by alternative pointing devices such as trackballs, head tracking, or gaze interaction. Trackballs were used in several text entry methods for a gestural input [42, 144] or on-screen keyboards (e.g., SpreadKey [82]). Head tracking is a popular mouse substitute for quadriplegic people, and many commercial products already exist (e.g., SmartNav by NaturalPoint[3]).

5.2 Gaze interaction

Gaze interaction is based on an eye tracker device which computes the focal point by measuring eye positions. This interaction has been used in many virtual keyboards. The absence of an explicit selection technique in gaze interaction has been mostly addressed by the dwell-time approach or multimodal interaction (see Sect. 2.3 for further detail).

Using dwell time, typing was found slower than mouse though comparable to head tracking [36]. Dwell time can be adjusted to gain maximum speed [76], and feedback can be improved by visual or audio cues [75]. The accuracy of typing can be improved by prediction [71]. Dwell time, entry rate, error rate, and workload were analyzed in an extensive study by Räihä and Ovaska [106]. Špakov and Majaranta [123] presented a scrollable keyboard which occluded minimal screen space. Panwar et al. [98] described an optimized design based on Fitts' law.

The Midas touch problem was solved by Isokoski [41] by using off-screen targets adjacent to the screen. Dwell time can be avoided either by context switching between two on-screen keyboards [89] or by using prediction [51].

The selection by other modalities includes selection by smiling [134], tooth clicking [151], or speech recognition [8], in which users pronounce a letter and look at it at the same time. Eye trackers are usually very expensive; however, several low-cost devices exist which can be used for text input [7, 111]. Gaze gestures were used, for example, in EdgeWrite [147] or Dasher [133]. Urbina and Huckauf [135] presented selection by gaze gesture in hierarchical pie menus.

5.3 Acoustic modality

Acoustic input modality can also be used for entering text. Many automatic speech recognition (ASR) systems support entering text by dictation [95] or letter by letter (e.g., [77, 78, 94]). An interesting combination of ASR and continuous gesturing was presented in Speech Dasher [136]. Motor-impaired people reach similar performance to able-bodied people with an ASR system [114]. However, users with speech impairments (e.g., dysarthria) report poor accuracy of ASR systems. Sometimes, it can be partially addressed by tedious training of an ASR system or by customizing keywords [34]. The accuracy still remains lower, however, when compared to people with clear speech.

For people with combined motor and speech impairment, one solution is *nonverbal vocal interaction* [125]. In this interaction modality, sounds other than speech are used, such as whistling, humming, and hissing. Several text entry methods for the nonverbal vocal interaction exist. Sporka et al. [124] describe three text entry methods operated by humming based on direct selection by tone pitch, Morse code, and inflected tones unique to each letter. CHANTI [126] is an ambiguous keyboard with word-level disambiguation on three keys. The keys are selected directly by one of four humming commands. Humsher

[2] http://www.maltron.com/.

[3] http://www.naturalpoint.com/smartnav/products/4-at/.

[101, 104] is a predictive keyboard which combines scanning and direct selection based on the ability of the user. Hissing was used as a switch for a scanning keyboard [103].

ASR systems often put users' privacy at risk. The users need to vocalize their input. ASR systems are also prone to recognition errors when a regional accent of English is used.

5.4 Biosignals

Another technique, which can be used in text input, is the measurement of biosignals. Two main noninvasive modalities belong to this technique—*intentional muscle contractions* and *brain–computer interaction*. Intentional muscle contractions are measured as peaks in an EMG (electromyography) signal which is produced by skeletal muscles. The electrodes are attached on the user's skin. For example, Felzer et al. [20, 25] attached an electrode to the forehead, and the switch was triggered by raising the eyebrow.

In the context of brain–computer interaction, EEG (electroencephalography) signals recorded along the scalp are often used. The process of decision making in the human brain yields a P300 wave [81]. When the P300 wave is detected, a switch is triggered. EEG uses more expensive hardware than EMG, and signal processing is more complex though it can help paralyzed users with a locked-in syndrome.

Text input based on EEG was used in P300 speller [139]. The keyboard is a 6×6 matrix containing alphanumeric characters. The user focuses on a character, and as the character flashes, the brain produces a stimulus. By design, at least two flashes are needed to input a character. In RSVP [39, 97], the letters are presented one after another, sorted according to the probability in a letter-level language model. The speed of these keyboards is still quite slow, and 1.5 WPM for P300 speller and 2 WPM for the RSVP were reported.

6 Experimental setup of text entry method evaluations

Every new text entry method has to be thoroughly evaluated to assess its possibilities and limitations. The experiments described in the literature often vary greatly in methodology, design, and evaluation. No standardized experimental setup exists except the one described in ISO 9241-4, which is designed for evaluation of a standard keyboard and not a common text entry method. Therefore, almost no paper surveyed this experiment. A number of best practices for the evaluation of text entry methods are described in the work by MacKenzie [65].

A common feature of text entry methods is relatively slow learning by the users. For example, learning a new PC keyboard layout takes more than 100 h [117]. Thus, it is quite common that the evaluation is spread into several sessions to grasp at least the beginning of the learning process. The speed of expert and novice users usually varies significantly.

6.1 Basic experimental setups

According to the literature review, five basic experimental setups are reported most often—simulation, evaluation with able-bodied users, evaluation with the target group, longitudinal evaluation, and expert evaluation.

Simulation is referred to as a process in which user input is replaced by an algorithm (e.g., [63]). The algorithm simulates user interaction with the examined method by computing a theoretical performance value, such as the number of keystrokes required to enter a portion of text. Simulation may be used for theoretical comparison among several layouts in terms of keystroke saving rate (KSR, e.g., [13]). From these values, the type rate can be estimated (e.g., [84, 104]). Simulation is essential for computing an optimal letter layout (see Sect. 3.1). Evaluation by simulation is relatively fast and is convenient for initial design comparisons and assessments. However, it does not take into account human factors such as immediate usability, errors produced, number of foci of attentions, visual search demands, and interaction demands. Simulation is error free and thus models the performance of an expert rather than a novice user.

Evaluation with able-bodied users is more accurate compared to simulation as it takes into account some human factors. This experimental setup is usually used for comparisons among several designs. This experimental setup often yields quantitative and qualitative results. However, neither qualitative nor quantitative results can be generalized for motor-impaired people. Qualitative results lack the necessary insight, and it is not guaranteed that the examined method posed an asset for them. Quantitative results are often significantly different in terms of type rate. For example, Vigouroux et al. [137] found that participants with spinal muscular atrophy are approximately 40–50 % slower than able-bodied when typing on an on-screen keyboard. On the other hand, evaluation with able-bodied people can still be used as a proof of concept [67].

Evaluation with the target group is used as a validation of the studied text entry method for the target group. It is mostly carried out in the form of case studies. The results are mostly qualitative providing a thorough insight. The reported case studies include only several participants as the study organization with the target group is usually too expensive due to challenging recruitment and transportation difficulties. This is one of the reasons why this kind of evaluation is missing in many scientific papers.

Longitudinal evaluation refers to an evaluation assessing performance in number of sessions (usually 10 and more). This type of evaluation gives the opportunity to model the performance according to power law of practice [91] and predicts the performance of an experienced user.

Expert evaluation is a short report of peak performance of evaluated text entry method. The expert is an experienced user of the method though in some cases the designation "expert" is attributed to a person with limited experience with the method. The performance of an expert user is usually reported in order to express an approximation of upper limit of the method's type rate.

The setups listed above usually provide different results. For example, the type rate of an expert user is much faster than that obtained in a case study with a disabled participant. However, the reported values are affected by other variables such as instructions given to the participants, length and number of session, choice of phrases to copy, and possibility of error correction.

6.2 Measures

While type rate is considered as the most important measure, a number of measures expressing other properties of the examined method are reported in the literature. Most of them are described in the work by Wobbrock [143]. Measures can be classified from several points of view:

- *Aggregate/character level.* Aggregate measures summarize the performance of a method as a whole. The most common are type rates, error rates, and efficiency measures. Character-level measures express performance for each character (e.g., letter confusion matrix).
- *Method agnostic/method specific.* According to [143], method-agnostic measures can be used for a variety of methods, while method-specific measures for one class of text entry methods only.
- *Absolute/relative.* Relative measures express a relation between two methods, for example, keystroke saving rate (KSR) [13] or selection savings [16]. Relative measures are used for comparisons. Absolute measures can be used to express performance of one method independently.
- *Empirical/theoretical.* Empirical values are measured in an experiment with people, while theoretical values are products of simulation, analysis, or estimation (e.g., KSPC [62]). Theoretical values usually do not take into account errors produced by the user. Theoretical and empirical values differ mostly for novice users. For example, the reported values of KSPC for MultiTab method [99] are 2.0 (theoretical) and 2.2 (empirical).
- *Subjective/objective.* Objective data measure performance of the text entry method, while subjective data express opinion of the users. Subjective data can be further divided to qualitative and quantitative. Qualitative data regard to immediate usability [69], participant comments, and researcher observations during the evaluation. Subjective quantitative data are usually obtained from a Likert scale involving several Likert items [57].

The *type rate* is mostly reported in terms of words per minute (WPM) where a "standard word" is defined as a string of 5 characters, including spaces. Other similar metrics are characters per minute (CPM), characters per second (CPS; $CPS = 5 * CPM$), or seconds per character (SPC; $SPC = 1/CPS$).

Reporting *error rate* is much less standardized. Three favorite error rates exist—keystrokes per character (KSPC) [62], MSD error rate [121], and a unified error metric [122]. KSPC expresses errors that were corrected during the experiment, while MSD corresponds to errors left in the transcribed text. The unified error metric provides similar measures: uncorrected and corrected error rate. The advantage of the unified error metric is the capability to report total error rate as the sum of uncorrected and corrected rates. KSPC and MSD error rates can not be combined this way [122].

Several other error rate metrics are reported in the literature as well: wrong characters rate (e.g., [80], see Eq. 1), overproduction rate (e.g., [35], see Equ. 2), or word error rate [136]. Character-level errors are usually presented by a confusion matrix (e.g., [18]).

$$wrong\ character\ rate = \frac{\#\ of\ wrong\ characters}{\#\ of\ total\ characters} \tag{1}$$

$$overproduction\ rate = \frac{\#\ of\ total\ keystrokes}{\#\ of\ optimal\ keystrokes} \tag{2}$$

7 Summary and evaluation of text entry methods

The text entry methods for motor-impaired people are summarized in Table 1. Each row corresponds to one scientific publication from year 1988 to year 2013.

7.1 Overview of target groups

As mentioned in the beginning of this paper, various causes of motor impairments exist, differing in the amount of impact on the users' physical ability to enter

Table 1 Summary of methods classified by **target group** (TG, see Sect. 7.1 for details), **selection technique** (D: direct, S: scanning, P: pointing), **character layout** (S: static, D: dynamic), **use of language model** (Pl: letter-level prediction, Pw: word-level prediction, Al: ambiguous keyboard with letter-level disambiguation, Aw: ambiguous keyboard with word-level disambiguation), **interaction modality** (see Sect. 7), **evaluation** (A: able-bodied participants, D: disabled participants, L: longitudinal, S: simulation), and **type rate** in terms of words per minute (WPM)

Year	Name and ref.	TG	Selection			Layout		Use of LM				Modality	Evaluation				WPM
1988	Chubon [14]	7	D			S						K(26)		A			
1988	*P300 speller* [19]	1		S		S						EEG		A			0.5
1992	*Reactive Kbd.* [15]	7	D			S			Pw			K(26)		A			
1993	*Half-QWERTY* [80]	7	D			S						K(13+1)		A		L	23–42
1998	Jones [45]	6		S		S		Pl	Pw			S		A			
1998	Kushler [53]	7	D			S					Aw	K(8)					
2000	*GRAFIS* [3]	2c		S	P	S			Pw			S/M					
2000	*MDITIM* [42]	6	D			S								A			3–10
2000	*Dasher* [140]	6			P		D	Pl				P/M/HT/G		A			18–34
2001	*UKO* [52]	7	D			S					Aw	K(4)			D		6
2001	*SUITEKeys* [78]	5	D			S			Pw			ASR					16
2002	*TouchMeKey4* [128]	7	D			S		Pl	Pw			K(4)	S	A			14–23
2003	*GazeTalk* [35]	3			P		D	Pl	Pw			P/M/G		A			4–6
2003	*UKO-II* [38]	7	D	S		S					Aw	K(?)					
2003	*EdgeWrite* [145]	6			P	S						TP		A	D		6–7
2004	Evreinova [18]	6	D			S						K(4)		A			15
2006	Baljko [6]	2c		S		S						K(1+1)	S	A			1.8
2006	*LURD-Writer* [21]	2b			P	S			Pw			EMG			D		1–2
2006	Sporka [124]	4	D			S						NVVI		A			2–3
2006	*EdgeWrite* [144]	6			P	S						TB		A	D		7–12
2007	*SEK* [48]	6			P	S						HT/M		A	D		2–3
2007	Lin [58]	6		S	P	S									D		
2007	*AUK* [90]	7	D	S		S						K(1-10)					
2007	Norte [93]	6		S	P	S						M/K(1)			D		
2008	*HandiGlyph* [9]	2c		S		S					Aw	PB			D	L	2–3
2008	Lin [59]	2c		S		S								A			1–2
2008	MacKenzie [71]	3			P	S		Pl	Pw			G		A			10–13
2008	Miro-Borras [83, 84]	2c		S			D			Al		PB	S				
2008	Spakov [123]	3			P	S						G		A			7–16
2008	*Dasher* [133]	3			P		D	Pl				G		A		L	15–20
2008	Sibylle [138]	2c		S		S		Pl	Pw			PB					
2008	*EyeWrite* [147]	3			P	S						G		A		L	2–8
2009	*3dScan* [22]	2b		S		S						EMG			D		1–2
2009	MacKenzie [66]	2c		S		S					Aw	K(1)		A			4–6
2009	Majaranta [76]	3			P	S						G		A		L	16–22
2010	*BlinkWrite2* [5]	2a		S		S					Aw	G		A			4–6
2010	*Qanti* [25]	2b		S		S					Aw	EMG		A	D		2–7
2010	*SAK* [67]	2b		S		S					Aw	K(1)/EMG	S	A	D		2–8
2010	*SpreadKey* [82]	6			P		D	Pl				TB	S		D	L	13
2010	Miro-Borras [86]	2c		S		S	D			Al	Aw	K(1)	S	A			6–16
2010	*KKBoard* [89]	3			P	S						G		A			10–20
2010	Roark [108]	2c		S			D	Pl				PB		A			2–5

Table 1 continued

Year	Name and ref.	TG	Selection			Layout		Use of LM			Modality	Evaluation				WPM
2010	Song [120]	6		S		S	D	Pl	Pw		J		A			5–8
2010	*pEYEwrite* [135]	3			P	S	D	Pl			G		A		L	10–17
2010	*Speech Dasher* [136]	5			P		D	Pl			ASR/G		A			40–54
2011	*CHANTI* [126]	4	D			S				Aw	NVVI			D		2–5
2011	*Humsher* [101]	4	D	S			D	Pl			NVVI		A	D		3–6
2011	*RSVP* [39]	1		S			D	Pl			EEG			D		
2011	*BlinkWrite* [113]	2a		S		S				Aw	G		A			4–5
2011	Prabhu [105]	2a		S		S					K(1)	S	A			1.3
2012	Beelders [8]	5			P	S					ASR/G		A			2–4
2012	*DualScribe* [26]	7	D			S				Aw	K(18)	S	A	D		3–5
2012	Lafi [54]	2c		S		S					K(1)					
2012	*Cont. EdgeWrite* [79]	6			P	S			Pw		GC		A			5
2012	*RSVP* [97]	1		S			D	Pl			EEG		A	D		1.8
2012	*EyeBoard* [98]	3			P	S					G		A			4–5
2012	Raiha [106]	3			P	S					G		A			20–24
2012	Zhao [151]	3			P	S					G		A			8–12
2012	Roark [109]	2c		S			D	Pl			K(1)		A			2–5
2013	Tuisku [134]	3			P	S					G		A			3–4

text. The extent of physical ability ranges from reduced dexterity of upper limbs to extreme conditions such as the locked-in syndrome. All publications cited in this paper have been clustered in the following categories which have been defined by the maximum extent of the physical ability of the target groups intended for the respective publications:

- (1) People with **locked-in** syndrome;
- (2) People capable of using **single switch interfaces**;
    - (2a) The switch is activated by **blinking**
    - (2b) The switch is activated by **facial muscle contractions**
    - (2c) The switch is a **push button**
- (3) People capable of using **eye tracker**;
- (4) People capable of **nonverbal vocal input**;
- (5) People capable of **speech**;
- (6) People capable of using **stationary pointing devices**, such as trackballs, joysticks, or 4-way arrows keyboards. Note: *Dasher* [140] is classified into this category even though a rather precise pointing is necessary to successfully use this method;
- (7) People with **reduced dexterity of upper limbs** using dedicated keyboards of reduced number of keys, including *Half-QWERTY* [80].

7.2 Summary of text entry methods

In order to select the methods presented in Table 1, a systematic literature review was conducted. In the first step, publications were collected by keyword search in the ACM Digital Library,[4] Springer Link,[5] IEEE Xplore,[6] and Google Scholar. The keywords were "text entry methods," "text input," "motor-impaired users," and "user study." Then, their references and citations were examined to add publications focusing on text input for motor-impaired people. This process was carried out recursively, until no new publication was added. From the obtained, pool of sixty-one were selected which reported on an original text entry method. The rest of publications were mostly studies of already existing methods.

The type rate in WPM was reported in most evaluations (49 out of 61 publications), and researchers are often compelled to use this characteristic as the only metric for comparison of different text entry methods. Other aspects of the studies should also be considered, not only because of conditions in the experimental setup (characteristics of the participants, number and length of sessions, instructions presented to the participants, phrases to copy, etc.) but also

[4] http://dl.acm.org/.

[5] http://link.springer.com/.

[6] http://ieeexplore.ieee.org/.

because of the way researchers report the type rate. Publications often state three different values. The most common is the arithmetic mean measured in the last session. Another one is the grand mean which equals to arithmetic mean measured across all session. Some publications also report the peak type rate achieved. Although all these values are used to express the type rate of the method, they are obviously quite different. Therefore, Table 1 does not show only one WPM value. It rather gives a range of WPM values expressing lowest and highest WPM values as reported in publications. The methods are categorized according to the primary target groups (see Sect. 7.1).

In the text entry methods summarized for motor-impaired people, selection by scanning is used in 27 of them, pointing in 24, and direct selection in 16. Note that the total number does not equal 61 as several papers describe more than one method design and those designs can have different properties. Unsurprisingly, scanning is recorded as the slowest input technique, as only one switch is used (WPM mean $= 3.75 \pm 2.5$, median $= 3.5$) when compared to average method using direct input (WPM mean $= 10.1 \pm 9.5$, median $= 6$) or pointing (WPM mean $= 12.0 \pm 10.5$, median $= 10$).

Scanning ambiguous keyboards appear to be the fastest among scanning methods. The aggregated WPM values for scanning ambiguous keyboards are WPM mean $=$ 5.3 ± 2.6 and median $= 5$, while the rest of the simple scanning methods yield WPM mean $= 2.6 \pm 1.7$ and median $= 1.8$.

Static layouts are highly preferred (50 publications) to the dynamic layouts (14). However, the authors did not find any trend in terms of type rate when comparing dynamic and static layouts.

Regarding the use of language models, prediction was used in 16 methods on the letter level and 12 methods on the word level. The disambiguation is popular on word level (11 publications), while letter-level disambiguation is reported only in two methods.

Modalities used in the methods are summarized in Fig. 17. Note that most methods use some kind of keyboard or keypad, usually restricted to only several buttons or keys. A large number of papers do not describe the actual interaction modality and specify only number of inputs.

7.3 Evaluation of text entry methods

Evaluation by simulation has been carried out in nine publications. An evaluation with able-bodied or disabled users is present in 53 out of 61 publications. The average numbers of participant count, session count, and session length are summarized in Table 2 separately for able-bodied and disabled participants. These number are shown together with the number of publications that contained the required information.

Note that evaluations with disabled participants were carried out in much fewer cases (40 vs. 18) and with fewer number of participants. Average number of sessions is similar in both cases, but session length is somewhat longer in case of disabled participants. Those numbers correspond to the premise that evaluation is usually quantitative with able-bodied participants and qualitative with disabled participants as described in Sect. 6.1. Reported type rates of disabled participants (WPM mean $= 4.5 \pm 3.2$, median $= 4$) are slower than those of able-bodied (WPM mean $= 9.3 \pm 9.6$, median $= 5$). This is partly because publications with high-speed methods often lack evaluation with disabled participants (see Table 1), and partly because

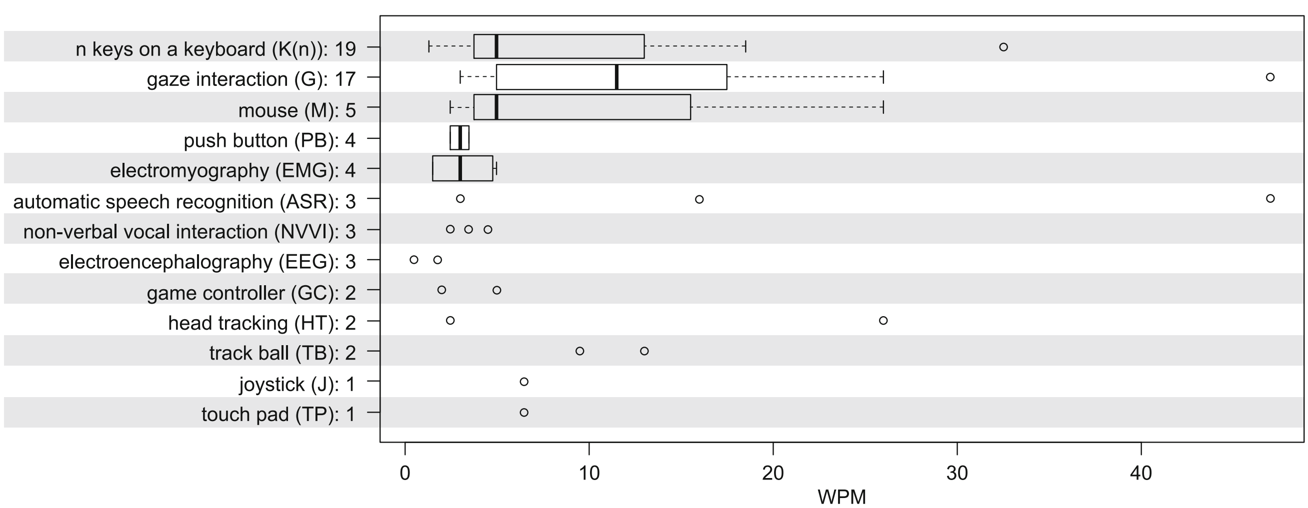


Fig. 17 Boxplot of WPM of the methods aggregated by modalities. Labels contain name of the modality, abbreviation, and the number of corresponding publications

Table 2 Summary of published evaluations of the methods

	No. of pub.	Mean (SD)	Median
Able-bodied participants			
Participants	40	8.8 (±4.0)	8.5
Sessions	39	5.3 (±4.7)	4
Session length	17	43 (±24.5)	50
Disabled participants			
Participants	18	2.5 (±2.7)	1
Sessions	11	5.4 (±4.6)	5
Session length	4	53 (±15)	60

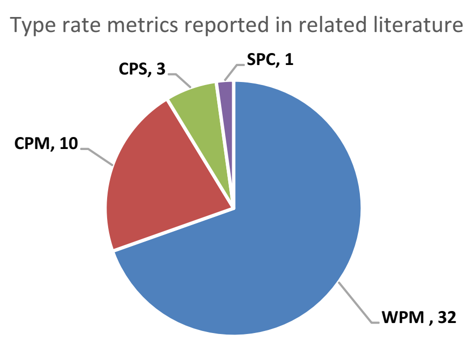


Fig. 18 Summary of type rate metrics as reported in the literature

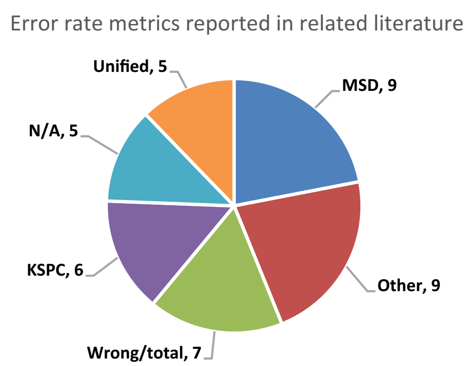


Fig. 19 Summary of error rate metrics as reported in the literature

they usually achieve slower type rates than able-bodied participants [137].

As shown in Fig. 18, the type rate is mostly reported in the word per minute (WPM) rate (32 evaluations) and characters per minute (CPM) rate (10). Other metrics of type rate are used only marginally. Note that slower methods are more likely to be reported in CPM. For CPM reports, the mean type rate is 3.5 ± 2.6 (median = 3.5), and for WPM the mean is 10.9 ± 10.0 (median = 6.5).

Error rate metrics are much more diverse as already mentioned in Sect. 6.1. Their use depends on the experiment setup. For example, MSD error rate (i.e., errors left in the transcribed text) is useless when error corrections are not allowed within the text-copy task. Figure 19 shows number of error rate metrics reported in the literature. The most widely used are MSD error rate (reported in nine publications), wrong characters rate (7), KSPC (6), and unified error rate (5). Nine publications report other metrics, such as overproduction rate, confusion matrix, or word error rate. In nine publications, error rate is reported, but the method of calculation is not stated.

Subjective data obtained by Likert items are reported in nine papers. They focus mostly on perceived speed, method/modality fatigue, ease of use, and efficiency. The full list of Likert items with number of occurrences in publications is shown in Table 3.

8 Conclusion

The aim of this paper is to fully describe the research related to text input for motor-impaired people. An overview of techniques for letter selection is given, and possibilities of letter distributions, prediction methods, and interaction modalities are discussed. The typical experimental setups and metrics which assess performance of a text entry method are also discussed. The methods found in the related scientific literature are summarized and classified according to the aspects listed above. The approximate type rate of each method is also shown, and implications are discussed.

Text entry for motor-impaired people is much slower than typing on a PC keyboard by an able-bodied person. The extreme case is scanning, which is used when only one input is available. Scanning ambiguous keyboards shows the best performance among the scanning text input. Although static layouts are highly preferred in the literature, their speed is comparable to dynamic layouts. Some dynamic layouts even outperform the static ones on the expense of escalated cognitive demands. As the text input for motor-impaired people is slow, prediction is popular technique to improve the text entry rate.

Table 3 Summary of Likert items reported in the literature

No. of pub.	Likert item
3	Speed, method fatigue
2	Ease of use, modality fatigue, efficiency
1	Stress, frustration, comfort, satisfaction, difficulty, accuracy, enjoyableness

Evaluation of text entry methods is still mostly carried out with able-bodied users, as a proof of concept. However, disabled participants usually achieve slower type rates in the experiments. The type rate is mostly reported in words per minute (WPM), though slower methods are more likely to be reported in characters per minute (CPM). No consensus exists on error rate metrics. Subjective data are obtained in qualitative or quantitative manner. Popular are Likert items focusing mostly on perceived speed, method/modality fatigue, ease of use, and efficiency.

Comparing different text entry methods is rather complicated as the experimental setup varies in each publication. Although ISO 9241-4 describes a standardized experimental setup, no reported evaluation follows this setup. It is argued that this is the case because the ISO 9241-4 experiment setup is tailored for manufacturers of computer keyboards.

A valid question arises whether a need exists for a standardized experimental setup which would apply to a general text entry method as a future work. Defining such setup should be a result of a broad academic discussion as it is necessary to find the trade-off between resources required and the validity of results. In order to improve the comparability of the methods, a standardized experimental setup should define namely the following:

- *Procedure,* including minimal number of participants, number of sessions, and length of one session or rather length of each experimental condition in one session. These three numbers should be balanced in order to yield a reasonable amount of hours to conduct the experiment.
- *Apparatus,* including phrases to copy, error correction capability, and error feedback if corrections are allowed. Participant instructions should also be standardized.
- *Dependent variables,* including type rate and error rate reporting. Standardized posttest questionnaires should be also developed.

On the other hand, the number of conditions should not be defined as often more methods are needed to be tested. However, general guidelines can be applied such as choice of the correct baseline, keeping the number of conditions low and ensuring correct counterbalancing.

When looking on the basic experimental setups, a potential in standardizing two of them is evident—the evaluation with able-bodied people and the longitudinal evaluation. Simulation is often too much dependent on the entry method to be standardized, and expert evaluation is reported only rarely. Moreover, such a potential cannot be seen in the standardization of an evaluation with disabled participants. The first reason is that the group of motor-impaired people is too diverse and it is often difficult to find two people with the exactly same conditions. The other reason is that organizing a quantitative experiment with motor-impaired people would be too challenging and expensive. Therefore, qualitative case studies are preferred by the researchers.

Acknowledgments This research has been supported by the projects TextAble (LH12070; Ministry of Education, Youth and Sports of the Czech Rep.) and Automatically Generated UIs in Nomadic Applications (SGS10/290/OHK3/3T/13; FIS 10-802900).

References

1. Andres, R.O., Hartung, K.J.: Prediction of head movement time using Fitts' law. Hum. Factors **31**(6), 703–713 (1989)
2. Anson, D., George, S., Galup, R., Shea, B., Vetter, R.: Efficiency of the Chubon versus the QWERTY keyboard. Assist. Technol. **13**(1), 40–45 (2001)
3. Antona, M, Stephanidis, C.: An accessible word processor for disabled people. In: Proceedings of the 7th International Conference on Computers Helping People with Special Needs, Österreichische Computer Gesellschaft, ICCHP 2000, pp. 689–696 (2000)
4. Arnott, J.L.: Text entry in augmentative and alternative communication. In: Harbusch, K., Raiha, K.J., Tanaka-Ishii, K. (eds.) Efficient Text Entry, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany, no. 05382 in Dagstuhl Seminar Proceedings (2006)
5. Ashtiani, B., MacKenzie, I.S.: Blinkwrite2: an improved text entry method using eye blinks. In: Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications, ETRA '10, pp. 339–345. ACM, New York (2010)
6. Baljko, M., Tam, A.: Indirect text entry using one or two keys. In: Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility, Assets '06, pp. 18–25. ACM, New York (2006)
7. Barrett, M., San Agustin, J.: Performance evaluation of a low-cost gaze tracker for eye typing. In: Proceedings of the 5th Conference on Communication by Gaze Interaction, COGAIN '09, pp. 13–18 (2009)
8. Beelders, T.R., Blignaut, P.J.: Measuring the performance of gaze and speech for text input. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA '12, pp. 337–340. ACM, New York (2012)
9. Belatar, M., Poirier, F.: Text entry for mobile devices and users with severe motor impairments: handiglyph, a primitive shapes based onscreen keyboard. In: Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '08, pp. 209–216. ACM, New York (2008)
10. Betke, M., Gips, J., Fleming, P.: The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities. IEEE Trans. Neural Syst. Rehabil. Eng. **10**(1), 1–10 (2002)
11. Boissiere, P., Dours, D.: VITIPI: versatile interpretation of text input by persons with impairments. In: Proceedings of the 5th International Conference on Computers Helping People with Special Needs. Part I, ICCHP '96, pp. 165–172. R. Oldenbourg Verlag GmbH, Munich, Germany (1996)
12. Boissiere, P., Dours, D.: An overview of existing writing assistance systems. In: Proceedings of the IFRATH Workshop 2003 (2003)

13. Boissiere, P., Vigouroux, N., Mojahid, M., Vella, F.: Adaptation of AAC to the context communication: a real improvement for the user illustration through the VITIPI word completion. In: Miesenberger, K., Karshmer, A., Penaz, P., Zagler, W. (eds.) Computers Helping People with Special Needs. Lecture Notes in Computer Science, vol. 7383, pp. 451–458. Springer, Berlin (2012)
14. Chubon, R.A., Hester, M.R.: An enhanced standard computer keyboard system for single-finger and typing-stick typing. Rehabil. Res. Dev. **25**(4), 17–24 (1988)
15. Darragh, J., Witten, I.H., James, M.: The reactive keyboard: a predictive typing aid. Computer **23**(11), 41–49 (1990)
16. Demasco, P.W., McCoy, K.F.: Generating text from compressed input: an intelligent interface for people with severe motor impairments. Commun. ACM **35**(5), 68–78 (1992)
17. Dvorak, A., Merrick, N., Dealey, W., Ford, G.: Typewriting Behavior: Psychology Applied to Teaching and Learning Typewriting. American Book Company, USA (1936)
18. Evreinova, T., Evreinov, G., Raisamo, R.: Four-key text entry for physically challenged people. In: Adjunct Proceedings of the 8th ERCIM Workshop User Interfaces For All, UI4ALL 04 (2004)
19. Farwell, L., Donchin, E.: Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalogr. Clin. Neurophysiol. **70**(6), 510–523 (1988)
20. Felzer, T., Freisleben, B.: Hawcos: the "hands-free" wheelchair control system. In: Proceedings of the Fifth International ACM Conference on Assistive Technologies, Assets '02, pp. 127–134. ACM, New York (2002)
21. Felzer, T., Nordmann, R.: Alternative text entry using different input methods. In: Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility, Assets '06, pp. 10–17. ACM, New York (2006)
22. Felzer, T., Rinderknecht, S.: 3dscan: an environment control system supporting persons with severe motor impairments. In: Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility, Assets '09, pp. 213–214. ACM, New York (2009)
23. Felzer, T., Rinderknecht, S.: Using a game controller for text entry to address abilities and disabilities specific to persons with neuromuscular diseases. In: The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '11, pp. 299–300. ACM, New York (2011)
24. Felzer, T., Strah, B., Nordmann, R.: Automatic and self-paced scanning for alternative text entry. In: Proceedings of the IASTED International Conference on Telehealth/Assistive Technologies, Telehealth/AT '08, pp. 1–6. ACTA Press, Anaheim (2008)
25. Felzer, T., MacKenzie, I., Beckerle, P., Rinderknecht, S.: Qanti: A software tool for quick ambiguous non-standard text input. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) Computers Helping People with Special Needs. Lecture Notes in Computer Science, vol. 6180, pp. 128–135. Springer, Berlin (2010)
26. Felzer, T., MacKenzie, I., Rinderknecht, S.: DualScribe: a keyboard replacement for those with Friedreichs Ataxia and related diseases. In: Miesenberger, K., Karshmer, A., Penaz, P., Zagler, W. (eds.) Computers Helping People with Special Needs. Lecture Notes in Computer Science, vol. 7383, pp. 431–438. Springer, Berlin (2012)
27. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. J. Exp. Psychol. **47**, 381–391 (1954)
28. Fu, Y., Huang, T.: hmouse: Head tracking driven virtual computer mouse. In: IEEE Workshop on Applications of Computer Vision, 2007, WACV '07, pp. 30–30 (2007)
29. Garay-Vitoria, N., Abascal, J.: Text prediction systems: a survey. Univ. Access Inf. Soc. **4**(3), 188–203 (2006)
30. Gong, J., Tarasewich, P.: Alphabetically constrained keypad designs for text entry on mobile devices. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '05, pp. 211–220. ACM, New York (2005)
31. Google: Ngram viewer. http://storage.googleapis.com/books/ngrams/books/datasetsv2.html (2013)
32. Grover, D.L., King, M.T., Kushler, C.A.: Reduced keyboard disambiguating computer. Technical Report, US Patent Publication (1998)
33. Gutowitz, H.: Barriers to adoption of dictionary-based text-entry methods: a field study. In: Proceedings of the 2003 EACL Workshop on Language Modeling for Text Entry Methods, TextEntry '03, pp. 33–41. Association for Computational Linguistics, Stroudsburg (2003)
34. Hamidi, F., Baljko, M., Livingston, N., Spalteholz, L.: Canspeak: A customizable speech interface for people with dysarthric speech. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) Computers Helping People with Special Needs. Lecture Notes in Computer Science, vol. 6179, pp. 605–612. Springer, Berlin (2010)
35. Hansen, J.P., Johansen, A.S., Hansen, D.W., Itoh, K., Mashino, S.: Language technology in a predictive, restricted on-screen keyboard with ambiguous layout for severely disabled people. In: Proceedings of EACL 2003 Workshop on Language Modeling for Text Entry Methods, EACL '03 (2003)
36. Hansen, J.P., Tørning, K., Johansen, A.S., Itoh, K., Aoki, H.: Gaze typing compared with input by head and hand. In: Proceedings of the 2004 Symposium on Eye Tracking Research and Applications, ETRA '04, pp. 131–138. ACM, New York (2004)
37. Harbusch, K., Kühn, M.: An evaluation study of two-button scanning with ambiguous keyboards. In: Proceedings of the 7th European Conference for the Advancement of Assistive Technology, AAATE'03, pp. 954–958 (2003)
38. Harbusch, K., Kühn, M.: Towards an adaptive communication aid with text input from ambiguous keyboards. In: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistic, EACL '03, vol. 2, pp. 207–210. Association for Computational Linguistics, Stroudsburg (2003)
39. Hild, K.E., Orhan, U., Erdogmus, D., Roark, B., Oken, B., Purwar, S., Nezamfar, H., Fried-Oken, M.: An erp-based brain–computer interface for text entry using rapid serial visual presentation and language modeling. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations, HLT '11, pp. 38–43. Association for Computational Linguistics, Stroudsburg (2011)
40. Huffman, D.: A method for the construction of minimum-redundancy codes. Proc. IRE **40**(9), 1098–1101 (1952)
41. Isokoski, P.: Text input methods for eye trackers using offscreen targets. In: Proceedings of the 2000 Symposium on Eye Tracking Research and Applications, ETRA '00, pp. 15–21. ACM, New York (2000)
42. Isokoski, P., Raisamo, R.: Device independent text input: a rationale and an example. In: Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '00, pp. 76–83. ACM, New York (2000)
43. Jacob, R.J.K.: The use of eye movements in human-computer interaction techniques: what you look at is what you get. ACM Trans. Inf. Syst. **9**(2), 152–169 (1991)
44. Johansen, A., Hansen, J.: Augmentative and alternative communication: the future of text on the move. In: Carbonell, N., Stephanidis, C. (eds.) Universal Access Theoretical Perspectives, Practice, and Experience. Lecture Notes in Computer Science, vol. 2615, pp. 425–441. Springer, Berlin (2003)

45. Jones, P.E.: Virtual keyboard with scanning and augmented by prediction. In: Proceedings of the 2nd European Conference on Disability, Virtual Reality and Associated Technologies, The University of Reading, ECDVRAT ’98, pp. 45–51 (1998)
46. Judge, S., Friday, M.: Ambiguous keyboards for AAC. J. Assist. Technol. **5**(4), 249–256 (2011)
47. Kim, H., Kim, Y.H.: Optimal designs of ambiguous mobile keypad with alphabetical constraints. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO ’09, pp. 1931–1932. ACM, New York (2009)
48. Kjeldsen, R.: An on-screen keyboard for users with poor pointer control. In: Stephanidis, C. (ed.) Universal Access in Human–Computer Interaction. Applications and Services, Lecture Notes in Computer Science, vol. 4556, pp. 339–348. Springer, Berlin (2007)
49. Koester, H.H., Levine, S.P.: Learning and performance of able-bodied individuals using scanning systems with and without word prediction. Assist. Technol. **6**(1), 42–53 (1994) **(pMID: 10147209)**
50. Költringer, T., Isokoski, P., Grechenig, T.: Twostick: writing with a game controller. In: Proceedings of Graphics Interface 2007, GI ’07, pp. 103–110. ACM, New York (2007)
51. Kristensson, P.O., Vertanen, K.: The potential of dwell-free eye-typing for fast assistive gaze communication. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA ’12, pp. 241–244. ACM, New York (2012)
52. Kühn, M., Garbe, J.: Predictive and highly ambiguous typing for a severely speech and motion impaired user. In: Universal Access in Human–Computer Interaction (2001)
53. Kushler, C.: AAC: Using a Reduced Keyboard (1998)
54. Lafi, S.M., Hock, S.K.B.: An adaptive text entry method based on single-key minimal scan matrix for people with severe motor disabilities. Sci. Eng. Res. **3**(8), 1–5 (2012)
55. Lesher, G., Moulton, B., Higginbotham, D.: Optimal character arrangements for ambiguous keyboards. IEEE Trans. Rehabil. Eng. **6**(4), 415–423 (1998a)
56. Lesher, G., Moulton, B., Higginbotham, D.J.: Techniques for augmenting scanning communication. Augment. Altern. Commun. **14**(2), 81–101 (1998b)
57. Likert, R.: A technique for the measurement of attitudes. Arch. Psychol. **22**(140), 5–55 (1932)
58. Lin, Y.L., Chen, M.C., Wu, Y.P., Yeh, Y.M., Wang, H.P.: A flexible on-screen keyboard: dynamically adapting for individuals needs. In: Stephanidis, C. (ed.) Universal Access in Human–Computer Interaction. Applications and Services, Lecture Notes in Computer Science, vol. 4556, pp. 371–379. Springer, Berlin (2007)
59. Lin, Y.L., Wu, T.F., Chen, M.C., Yeh, Y.M., Wang, H.P.: Designing a scanning on-screen keyboard for people with severe motor disabilities. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) Computers Helping People with Special Needs. Lecture Notes in Computer Science, vol. 5105, pp. 1184–1187. Springer, Berlin (2008)
60. Loewenich, F., Maire, F.: Hands-free mouse-pointer manipulation using motion-tracking and speech recognition. In: Proceedings of the 19th Australasian Conference on Computer–Human Interaction: Entertaining User Interfaces, OZCHI ’07, pp. 295–302. ACM, New York (2007)
61. Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., Looney, E.W. Twiddler typing: one-handed chording text entry for mobile phones. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’04, pp. 671–678. ACM, New York (2004)
62. MacKenzie, I.: KSPC (keystrokes per character) as a characteristic of text entry techniques. In: Patern, F. (ed.) Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science, vol. 2411, pp. 195–210. Springer, Berlin (2002)
63. MacKenzie, I.: Modeling text input for single-switch scanning. In: Miesenberger, K., Karshmer, A., Penaz, P., Zagler, W. (eds.) Computers Helping People with Special Needs. Lecture Notes in Computer Science, vol. 7383, pp. 423–430. Springer, Berlin (2012)
64. MacKenzie, I.S.: Movement time prediction in human–computer interfaces. Proc. Graph. Interface **92**, 140–150 (1992)
65. MacKenzie, I.S.: Evaluation of text entry techniques. In: MacKenzie, I.S., Tanaka-Ishii, K. (eds.) Text entry systems: mobility, accessibility, universality, pp. 75–101. Morgan Kaufmann (2007)
66. MacKenzie, I.S.: The one-key challenge: searching for a fast one-key text entry method. In: Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility, Assets ’09, pp. 91–98. ACM, New York (2009)
67. MacKenzie, I.S., Felzer, T.: Sak: Scanning ambiguous keyboard for efficient one-key text entry. ACM Trans. Comput. Hum. Interact. **17**(3), 11:1–11:39 (2010)
68. MacKenzie, I.S., Teather, R.J.: Fittstilt: the application of fitts’ law to tilt-based interaction. In: Proceedings of the 7th Nordic Conference on Human–Computer Interaction: Making Sense Through Design, NordiCHI ’12, pp. 568–577. ACM, New York (2012)
69. MacKenzie, I.S., Zhang, S.: The immediate usability of Graffiti. In: Proceedings of Graphics Interface, Canadian Information Processing Society, pp. 129–137 (1997)
70. MacKenzie, I.S., Zhang, S.X.: The design and evaluation of a high-performance soft keyboard. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’99, pp. 25–31. ACM, New York (1999)
71. MacKenzie, I.S., Zhang, X.: Eye typing using word and letter prediction and a fixation algorithm. In: Proceedings of the 2008 Symposium on Eye Tracking Research and Applications, ETRA ’08, pp. 55–58. ACM, New York (2008)
72. MacKenzie, I.S., Kober, H., Smith, D., Jones, T., Skepner, E.: LetterWise: prefix-based disambiguation for mobile text input. In: Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology, UIST ’01, pp 111–120. ACM, New York (2001)
73. MacKenzie, I.S., Soukoreff, R.W., Helga, J.: 1 thumb, 4 buttons, 20 words per minute: design and evaluation of h4-writer. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST ’11, pp. 471–480. ACM, New York (2011)
74. Majaranta, P., Räihä, K.J.: Twenty years of eye typing: systems and design issues. In: Proceedings of the 2002 Symposium on Eye Tracking Research and Applications, ETRA ’02, pp. 15–22. ACM, New York (2002)
75. Majaranta, P., MacKenzie, I., Aula, A., Räihä, K.J.: Effects of feedback and dwell time on eye typing speed and accuracy. Univ. Access Inf. Soc. **5**(2), 199–208 (2006)
76. Majaranta, P., Ahola, U.K., Špakov, O.: Fast gaze typing with an adjustable dwell time. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’09, pp. 357–360. ACM, New York (2009)
77. Manaris, B., Harkreader, A.: SUITEKeys: a speech understanding interface for the motor-control challenged. In: Proceedings of the Third International ACM Conference on Assistive Technologies, Assets ’98, pp. 108–115. ACM, New York (1998)
78. Manaris, B., McCauley, R.: An intelligent interface for keyboard and mouse control—providing full access to PC functionality via speech. In: Proceedings of 14th International Florida AI Research Symposium, FLAIRS-01, pp. 182–188. AAAI Press (2001)

79. Martin, B., Isokoski, P., Karmann, G., Rollinger, T.: Continuous edgewrite: dictionary-based disambiguation instead of explicit segmentation by the user. In: Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI '12, pp. 357–364. ACM, New York (2012)
80. Matias, E., MacKenzie, I.S., Buxton, W.: Half-QWERTY: a one-handed keyboard facilitating skill transfer from QWERTY. In: Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, CHI '93, pp. 88–94. ACM, New York (1993)
81. Meixner, J.B., Rosenfeld, J.P.: Detecting knowledge of incidentally acquired, real-world memories using a p300-based concealed-information test
82. Merlin, B., Raynal, M.: Evaluation of spreadkey system with motor impaired users. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) Computers Helping People with Special Needs. Lecture Notes in Computer Science, vol. 6180, pp. 112–119. Springer, Berlin (2010)
83. Miró, J., Bernabeu, P.: Text entry system based on a minimal scan matrix for severely physically handicapped people. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) Computers Helping People with Special Needs. Lecture Notes in Computer Science, vol. 5105, pp. 1216–1219. Springer, Berlin (2008)
84. Miró-Borrás, J., Bernabeu-Soler, P.: E-everything For All: Text entry for people with severe motor disabilities. In: Proceedings of the Collaborative Electronic Communications and eCommerce Technology and Research Iberoamerica, 6th CollECTeR, pp. 1–7 (2008)
85. Miró-Borrás, J., Bernabeu-Soler, P.: Text entry in the e-commerce age: two proposals for the severely handicapped. J. Theor. Appl. Electron. Commer. Res. **4**(1), 101–112 (2009)
86. Miró-Borrás, J., Bernabeu-Soler, P., Llinares, R., Igual, J.: Evaluation of an ambiguous-keyboard prototype scanning-system with word and character disambiguation. In: Proceedings of the 24th BCS Interaction Specialist Group Conference, BCS '10, pp. 403–411. British Computer Society, Swinton (2010a)
87. Miró-Borrás, J., Bernabeu-Soler, P., Llinares, R., Igual, J.: A prototype scanning system with an ambiguous keyboard and a predictive disambiguation algorithm. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) Computers Helping People with Special Needs. Lecture Notes in Computer Science, vol. 6180, pp. 136–139. Springer, Berlin (2010b)
88. Molina, A.J., Rivera, O., Gómez, I., Merino, M., Ropero, J.: Comparison among ambiguous virtual keyboards for people with severe motor disabilities. Modern Eng. Res. **1**(2), 288–305 (2011)
89. Morimoto, C.H., Amir, A.: Context switching for fast key selection in text entry applications. In: Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications, ETRA '10, pp. 271–274. ACM, New York (2010)
90. Mourouzis, A., Boutsakis, E., Ntoa, S., Antona, M., Stephanidis, C.: An accessible and usable soft keyboard. In: Stephanidis, C. (ed.) Universal Access in Human–Computer Interaction. Ambient Interaction, Lecture Notes in Computer Science, vol. 4555, pp. 961–970. Springer, Berlin (2007)
91. Newell, A., Rosenbloom, P.S.: Mechanisms of skill acquisition and the law of practice. In: Anderson, J.R. (ed.) Cognitive Skills and Their Acquisition, pp. 1–55. Lawrence Erlbaum Associates, Mahwah (1981)
92. Norman, D.A., Fisher, D.: Why alphabetic keyboards are not easy to use: keyboard layout doesn't much matter. Hum. Factors J. Hum. Factors Ergon. Soc. **24**(5), 509–519 (1982)
93. Norte, S., Lobo, F.G.: A virtual logo keyboard for people with motor disabilities. In: Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE '07, pp. 111–115. ACM, New York (2007)
94. Nouza, J., Nouza, T., erva, P.: A multi-functional voice-control aid for disabled persons. In: Proceedings of the 10th International Conference on Speech and Computer, Patras, Greece, SPECOM '05, pp. 715–718 (2005)
95. Nouza, J., Zdansky, J., Cerva, P., Silovsky, J.: Challenges in speech processing of slavic languages (case studies in speech recognition of Czech and Slovak). In: Esposito, A., Campbell, N., Vogel, C., Hussain, A., Nijholt, A. (eds.) Development of Multimodal Interfaces: Active Listening and Synchrony. Lecture Notes in Computer Science, vol. 5967, pp. 225–241. Springer, Berlin (2010)
96. Ntoa, S., Margetis, G., Antona, M., Stephanidis, C.: Scanning-based interaction techniques for motor impaired users. In: Kouroupetroglou, G. (eds.) Assistive Technologies and Computer Access for Motor Disabilities, IGI Global, Chap 3 (2013)
97. Orhan, U., Hild, K., Erdogmus, D., Roark, B., Oken, B., Fried-Oken, M.: RSVP keyboard: An EEG based typing interface. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 645–648 (2012)
98. Panwar, P., Sarcar, S., Samanta, D.: Eyeboard: a fast and accurate eye gaze-based text entry system. In: 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI), pp. 1–8. IEEE (2012)
99. Pavlovych, A., Stuerzlinger, W.: Less-Tap: A fast and easy-to-learn text input technique for phones. In: Graphics Interface, pp. 97–104 (2003)
100. Perlin, K.: Quikwriting: continuous stylus-based text entry. In: Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology, UIST '98, pp 215–216. ACM, New York (1998)
101. Polacek, O., Mikovec, Z., Sporka, A.J., Slavik, P.: Humsher: a predictive keyboard operated by humming. In: The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '11, pp. 75–82. ACM, New York (2011)
102. Poláček, O., Míkovec, Z.: Hands free mouse: comparative study on mouse clicks controlled by humming. In: CHI '10 Extended Abstracts on Human Factors in Computing Systems, CHI EA '10, pp. 3769–3774. ACM, New York (2010)
103. Poláček, O., Míkovec, Z., Slavík, P.: Predictive scanning keyboard operated by hissing. In: Proceedings of the 2nd IASTED International Conference Assistive Technologies, IASTED, AT 2012, pp. 862–869 (2012a)
104. Poláček, O., Sporka, A.J., Míkovec, Z.l.: Measuring performance of a predictive keyboard operated by humming. In: Proceedings of the 13th International Conference on Computers Helping People with Special Needs—Volume Part II, ICCHP'12, pp. 467–474. Springer, Berlin (2012b)
105. Prabhu, V., Prasad, G.: Designing a virtual keyboard with multimodal access for people with disabilities. In: 2011 World Congress on Information and Communication Technologies (WICT), pp. 1133–1138 (2011)
106. Räihä, K.J., Ovaska, S.: An exploratory study of eye typing fundamentals: dwell time, text entry rate, errors, and workload. In: Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems, CHI '12, pp. 3001–3010. ACM, New York (2012)
107. Rivera, O., Molina, A., Gómez, I.M., Merino, M.: A study of two-inputs scanning methods to enhance the communication rate. In: Emiliani, P., Burzagli, L., Como, A., Gabbanini, F., Salminen, A.L. (eds.) Assistive Technology from Adapted Equipment to Inclusive Environments, AAATE 2009, vol. 25, pp. 132–137. IOS Press (2009)

108. Roark, B., de Villiers, J., Gibbons, C., Fried-Oken, M.: Scanning methods and language modeling for binary switch typing. In: Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies, SLPAT '10, pp. 28–36. Association for Computational Linguistics, Stroudsburg (2010)
109. Roark, B., Beckley, R., Gibbons, C., Fried-Oken, M.: Huffman scanning: using language models within fixed-grid keyboard emulation. Comput. Speech Lang. **27**(6), 1212–1234 (2013)
110. Ronzhin, A., Karpov, A.: Assistive multimodal system based on speech recognition and head tracking. In: Proceedings of 13th European Signal Processing Conference, EUSIPCO '05 (2005)
111. San Agustin, J., Skovsgaard, H., Hansen, J.P., Hansen, D.W.: Low-cost gaze interaction: ready to deliver the promises. In: CHI '09 Extended Abstracts on Human Factors in Computing Systems, CHI EA '09, pp. 4453–4458. ACM, New York (2009)
112. Schadle, I.: Sibyl: AAC System Using NLP Techniques. In: Miesenberger, K., Klaus, J., Zagler, W., Burger, D. (eds.) Computers Helping People with Special Needs. Lecture Notes in Computer Science, vol. 3118, pp. 1009–1015. Springer, Berlin (2004)
113. Scott MacKenzie, I., Ashtiani, B.: Blinkwrite: efficient text entry using eye blinks. Univ. Access Inf. Soc. **10**(1), 69–80 (2011)
114. Sears, A., Karat, C.M., Oseitutu, K., Karimullah, A., Feng, J.: Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. Univ. Access Inf. Soc. **1**(1), 4–15 (2001)
115. Shieber, S.M., Baker, E.: Abbreviated text input. In: Proceedings of the 8th International Conference on Intelligent User Interfaces, IUI '03, pp. 293–296. ACM, New York (2003)
116. Sibert, L.E., Jacob, R.J.K.: Evaluation of eye gaze interaction. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '00, pp. 281–288. ACM, New York (2000)
117. Silfverberg, M.: Historical overview of consumer text entry technologies. In: MacKenzie, I.S., Tanaka-Ishii, K. (eds.) Text Entry Systems: Mobility, Accessibility, Universality, pp. 3–26. Morgan Kaufmann, Los Altos (2007)
118. Smith, B.A., Zhai, S.: Optimised virtual keyboards with and without alphabetical ordering: a novice user study. In: Hirose, M. (ed.) Proceedings of Interact 2001IFIP International Conference on HumanComputer Interaction, INTERACT'01, pp. 92–99. IOS Press (2001)
119. Solutions, T.: The fitaly one-finger keyboard (1998)
120. Song, Y.C.: Joystick text entry with word prediction for people with motor impairments. In: Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '10, pp. 321–322. ACM, New York (2010)
121. Soukoreff, R.W., MacKenzie, I.S.: Measuring errors in text entry tasks: an application of the Levenshtein string distance statistic. In: CHI '01 Extended Abstracts on Human Factors in Computing Systems, CHI EA '01, pp. 319–320. ACM, New York (2001)
122. Soukoreff, R.W., MacKenzie, I.S.: Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '03, pp. 113–120. ACM, New York (2003)
123. Špakov, O., Majaranta, P.: Scrollable keyboards for eye typing. In: Proceedings of the 4th Conference on Communication by Gaze Interaction, COGAIN '08, pp. 63–66 (2008)
124. Sporka, A., Kurniawan, S., Slavk, P.: Non-speech operated emulation of keyboard. In: Clarkson, J., Langdon, P., Robinson, P. (eds.) Designing Accessible Technology, pp. 145–154. Springer, London (2006)
125. Sporka, A.J.: Non-speech sounds for user interface control. Ph.D. thesis, CTU in Prague (2008)
126. Sporka, A.J., Felzer, T., Kurniawan, S.H., Poláček, O., Haiduk, P., MacKenzie, I.S.: Chanti: predictive text entry using non-verbal vocal input. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, pp. 2463–2472. ACM, New York (2011)
127. Tanaka-Ishii, K., Frank, I.: Dit4dah: Predictive pruning for morse code text entry. In: Su, K.Y., Tsujii, J., Lee, J.H., Kwong, O. (eds.) Natural Language Processing IJCNLP 2004. Lecture Notes in Computer Science, vol. 3248, pp. 765–775. Springer, Berlin (2005)
128. Tanaka-Ishii, K., Inutsuka, Y., Takeichi, M.: Entering text with a four-button device. In: Proceedings of the 19th International Conference on Computational Linguistics, COLING '02, vol. 1, pp. 1–7. Association for Computational Linguistics, Stroudsburg (2002)
129. Teahan, W.: Probability estimation for PPM. In: In Proceedings NZCSRSC'95. http://www.cs.waikato.ac.nz/wjt (1995)
130. Tregoubov, M., Birbaumer, N.: On the building of binary spelling interfaces for augmentative communication. IEEE Trans. Biomed. Eng. **52**(2), 300–305 (2005)
131. Trewin, S., Arnott, J.: Text entry when movement is impaired. In: MacKenzie, I.S., Tanaka-Ishii, K. (eds.) Text Entry Systems: Mobility, Accessibility, Universality, pp. 289–304. Morgan Kaufmann (2007)
132. Tu, J., Tao, H., Huang, T.: Face as mouse through visual face tracking. Comput. Vis. Image Underst. **108**(1–2), 35–40 (2007)
133. Tuisku, O., Majaranta, P., Isokoski, P., Räihä, K.J.: Now dasher! dash away!: longitudinal study of fast text entry by eye gaze. In: Proceedings of the 2008 Symposium on Eye Tracking Research and Applications, ETRA '08, pp. 19–26. ACM, New York (2008)
134. Tuisku, O., Surakka, V., Rantanen, V., Vanhala, T., Lekkala, J.: Text entry by gazing and smiling. Adv. Hum.–Comput. Interact. (2013)
135. Urbina, M.H., Huckauf, A.: Alternatives to single character entry and dwell time selection on eye typing. In: Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications, ETRA '10, pp. 315–322. ACM, New York (2010)
136. Vertanen, K., MacKay, D.J.: Speech dasher: fast writing using speech and gaze. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10, pp. 595–598. ACM, New York (2010)
137. Vigouroux, N., Vella, F., Truillet, P., Raynal, M.: Evaluation od AAC for text input by two groups of subjects: able-bodied subjects and disabled motor subjects. In: Adjunct Proceedings of the 8th ERCIM Workshop User Interfaces For All, UI4ALL 04 (2004)
138. Wandmacher, T., Antoine, J.Y., Poirier, F., Départe, J.P.: Sibylle, an assistive communication system adapting to the context and its user. ACM Trans. Access Comput. **1**(1), 6:1–6:30 (2008)
139. Wang, C., Guan, C., Zhang, H.: P300 brain–computer interface design for communication and control applications. In: IEEE-EMBS 2005. 27th Annual International Conference of the Engineering in Medicine and Biology Society, pp. 5400–5403 (2005)
140. Ward, D.J., Blackwell, A.F., MacKay, D.J.C.: Dasher—a data entry interface using continuous gestures and language models. In: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology, UIST '00, pp. 129–137. ACM, New York (2000)

141. Wilson, A.D., Agrawala, M.: Text entry using a dual joystick game controller. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’06, pp. 475–478. ACM, New York (2006)
142. Witten, I.H., Bell, T.: The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. IEEE Trans. Inf. Theory **37**(4), 1085–1094 (1991)
143. Wobbrock, J.: Measures of text entry performance. In: MacKenzie, I.S., Tanaka-Ishii, K. (eds.) Text Entry Systems: Mobility, Accessibility, Universality, Morgan Kaufmann, pp. 47–74 (2007)
144. Wobbrock, J., Myers, B.: Trackball text entry for people with motor impairments. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’06, pp. 479–488. ACM, New York (2006)
145. Wobbrock, J.O., Myers, B.A., Kembel, J.A.: EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In: Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology, UIST ’03, pp 61–70. ACM, New York (2003)
146. Wobbrock, J.O., Myers, B.A., Aung, H.H.: Writing with a joystick: a comparison of date stamp, selection keyboard, and EdgeWrite. In: Proceedings of Graphics Interface 2004, Canadian Human–Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, GI ’04, pp. 1–8 (2004)
147. Wobbrock, J.O., Rubinstein, J., Sawyer, M.W., Duchowski, A.T.: Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In: Proceedings of the 2008 Symposium on Eye Tracking Research and Applications, ETRA ’08, pp. 11–18. ACM, New York (2008)
148. Yin, P.Y., Su, E.P.: Optimal character arrangement for ambiguous keyboards using a PSO-based algorithm. In: 2011 Seventh International Conference on Natural Computation (ICNC), vol. 4, pp. 2194–2198 (2011)
149. Zhai, S., Kristensson, P.O.: Shorthand writing on stylus keyboard. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’03, pp. 97–104. ACM, New York (2003)
150. Zhai, S., Hunter, M., Smith, B.A.: The metropolis keyboard—an exploration of quantitative techniques for virtual keyboard design. In: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology, UIST ’00, pp. 119–128. ACM, New York (2000)
151. Zhao, X.A., Guestrin, E.D., Sayenko, D., Simpson, T., Gauthier, M., Popovic, M.R.: Typing with eye-gaze and tooth-clicks. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA ’12, pp. 341–344. ACM, New York (2012)