

Evaluation of an Ambiguous-Keyboard Prototype Scanning-System with Word and Character Disambiguation

Julio Miró-Borrás Pablo Bernabeu-Soler Raul Llinares Jorge Igual
Universidad Politécnica de Valencia, Departamento de Comunicaciones
Plaza Ferrandiz-Carbonell s/n, 03801 Alcoy (Spain)
jmirobo@dcom.upv.es pbernabe@dcom.upv.es rllinares@dcom.upv.es jigual@dcom.upv.es

Ambiguous keyboards are common in small-size devices such as mobile phones, but they can be applied to other fields such as Augmentative and Alternative Communications, and specifically addressed to people with severe motor disabilities. This research proposes a novel alternative to assign letters to keys, where letters with similar graphical characteristics are grouped into the same key, leading to families of four, three and two-key simple and easy-to-learn layouts. All layouts have been analyzed for a scanning system that implements a word and a character disambiguation algorithm using only one switch as input device. With the best three-key layout, the text entry speeds predicted were 16.7 wpm using word disambiguation and 10.8 wpm for character disambiguation, with a 0.5 seconds scan period. In an experiment conducted with 6 participants and dynamic scan period, average entry speeds reached 12.2 and 6.3 wpm respectively. The top speeds reached by single participants were 16.9 and 8.3 wpm.

Augmentative and Alternative Communications, text entry, ambiguous keyboards, disambiguation, scanning.

1. INTRODUCTION

Keyboards with fewer keys than characters (ambiguous keyboards) are common in small-size devices such as mobile phones, but they can be applied to other fields such as Augmentative and Alternative Communications (AAC), and specifically addressed to people with severe motor disabilities. The use of scanning systems with one switch as an input device makes it necessary to reduce the scan period in order to increase the text entry speed in wpm (words per minute). With short scan periods, users have little time to decide if the desired letter is included in the cell just highlighted. In this way, benefits may be obtained when using simple and easy-to-learn layouts.

This research proposes a novel alternative to assigning letters to keys called CGA (Character Graphical Association), where letters with similar graphical characteristics are grouped into the same key, leading to families of four, three and two-key simple and easy-to-learn keyboards. All CGA layouts were analyzed when assigned to a text entry system based on scanning with the two disambiguation algorithms described in [17]: Word Level Disambiguation Mode (WLDM) and Character Level Disambiguation Mode (CLDM). The scanning system uses only one switch as input device. We calculated and compared the performance of all CGA layouts using both disambiguation modes (DM) and then selected the best one: CGA3-122¹. Then

we conducted an experiment to compare CGA3-122 with both DM, in order to check if the theoretical performance is reasonable in practice, and made comparisons on text entry speed, scan periods, and improvement with practice. Finally, in the experiment, we also estimated a figure of merit that comprises both the scan period and the accuracy.

The main literature related to ambiguous keyboards and text entry systems based on scanning is reviewed in the subsections below. The remainder of this paper is organized as follows. In the following section we propose the CGA specifications. Then, we quickly review the methodology for analyzing scanning systems with ambiguous keyboards, specifically those working with WLDM and CLDM [17], [18]. We also present an empirical evaluation of a scanning system with layout CGA3-122 and both DM. In the results and discussion section, we present both the analytical data and the empirical results of the experiment, followed by the main conclusions.

¹ CGA3-122 is the given name for the 122th layout in our list of 3-key family of keyboards.

2. BACKGROUND

2.1 Layouts Variants

The standard phone keyboard has evolved towards other layout variants with different numbers of keys and different assignments of letters to keys. As [13] summarizes, these layouts may follow: (a) “abc” ordering, e.g., standard phone, ACD, L6K, L4K, L2K, TM4, etc.; (b) “QWERTY” ordering such as QP10213, Suretype, EQ6, EQ3 or even Stick; and (c) “optimized” ordering such as AKKO, or LetterEase. Both “abc” and “QWERTY” orderings try to take advantage of letter distribution knowledge. Optimized ordering tries to obtain higher text entry speeds, although users need more time to learn the layout. Recently, a new approach named UniGlyph [1] is based on the shape of capital Latin characters, grouping them according to some graphical primitives.

The number of keys with letters in ambiguous keyboards typically ranges from 18 (EQ6) to only 2 (L2K), plus one key for “space” and another for the “next” function, needed in predictive disambiguation methods.

Ambiguous keyboards may be implemented with physical buttons (physical keyboards) but also on screens (virtual keyboards). In this case, keys can be accessed by pressing directly the screen or by using one or more switches as input devices and a scanning technique that moves the cursor among the keys [17]. This technique allows users with motor disabilities to enter text into a computer.

2.2 Disambiguation Methods

Having several letters on each key makes it necessary to use a procedure to indicate to the system the intended character or word the user wants to enter. Some predictive methods are one-key with disambiguation (e.g. T9 [6]) or LetterWise [16]. Some non predictive ones are multitap or two-keys, and are described in [13].

2.3 Text Entry Systems Based on Scanning

Text entry is done using virtual keyboards and a scanning technique. Scanning interfaces move the focus of control in a grid, sequentially and automatically from item to item, with a standard timeout between moves. The user only needs to press the switch to select the item when automatic scanning is used. In the case of inverse scanning, the selection is done simply by releasing the switch [21].

Traditional virtual keyboards used in scanning systems assigned one character to each cell, creating a scan matrix with a large number of cells. Most of the research in this area was done

in the 70's and 80's and several strategies were proposed to increase the text entry speed such as (a) rearrangement of the characters in the scan matrix, (b) grouped access to cells, typically R-C scanning (row-column), and (c) prediction of the next block (mainly characters, words or sentences). The typical speeds for one-switch systems ranged from 0.5 to 5 wpm as Darragh and Witten summarize in [5]. Although they tried to compare the proposals, data available from scanning systems was quite inaccurate, being difficult to make comparisons between systems.

Damper [4] presented a mathematical model in order to compare the text entry speed of scanning systems, and applied it to a 6x7 scan matrix with a scan period of 0.5 seconds. He predicted a text entry speed for this system of about 6.5 wpm. Leshner et al. [10] compared several layouts, with different sets of items and prediction strategies, in terms of keystroke savings, concluding that the best performance is obtained using an optimized configuration with a seven element character prediction list and a 7x7 elements scan matrix. No text entry in wpm was estimated. Higher text entry speeds can be obtained by reducing the scan period. A strategy to alter dynamically the scan period is described in [11].

Biswas and Robinson [2] developed a simulator that can predict possible interaction patterns when undertaking a task using a variety of input devices, and estimate the time to complete the task in the case of different disabilities and for different levels of skills.

Recently, ambiguous keyboards are being used as a replacement of the scan matrices in order to obtain a better performance. Kühn and Garbe presented a system called UKO with the letters mapped onto four keys plus two auxiliary keys [9]. They reported a text input speed of 6 wpm, using two switches by a 15-year old girl with cerebral palsy. Harbusch and Kühn [7] presented a study about five virtual keyboards, some of them ambiguous, with the results in scan steps per word, concluding the convenience of using ambiguous keyboards to obtain better results. Subsequently, they described an application called UKO-II with the letters mapped onto three keys plus an auxiliary key [8]. MacKenzie [14] proposed SAK (Scanning Ambiguous Keyboard) with three keys for letters arranged alphabetically plus an extra key for the “space”. He reported the following empirical results: 5.11 wpm (all trials, 99% accuracy) and 7.03 wpm (error-free trials).

The words “key” and “cell” are used interchangeably in this paper when referring to ambiguous keyboards in scanning systems.

All of the preceding proposals use disambiguation algorithms based on word prediction or word completion. In [18] we presented a two-key keyboard with CLDM, and in [17], we expanded it introducing

WLDM, using a three-key alphabetically ordered keyboard.

WLDM is an adaptation from the method “one key with disambiguation” from mobile phones and the disambiguation is done after selecting all the necessary cells to enter a word. It uses neither a separate key for “space” nor another one for the “next” function needed in the disambiguation process. Both of them are replaced with a combination of scanning modes: automatic and inverse scanning.

CLDM is an adaptation from the “multitap” method used in mobile phones, and is inspired by the LetterWise proposal [16]. There is no need to have a dedicated cell for the space because of the use of a combination of automatic and inverse scanning. In this mode, disambiguation is done letter by letter, so once a cell is selected, it is necessary to start a disambiguation process in order to choose the right letter included in the cell.

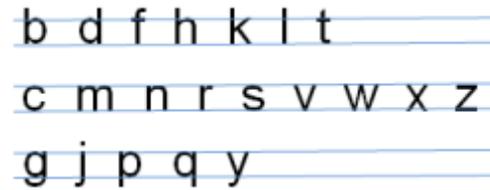
With respect to the number of keys, scanning systems with ambiguous keyboards perform better with reduced sets of keys [19].

3. CGA SPECIFICATIONS

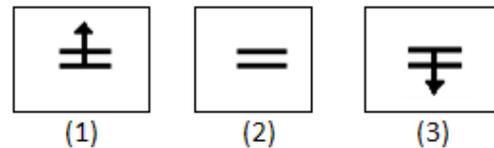
CGA represents a new way of assigning letters to keys and allows us to generate keyboard families with two, three and four keys. We group the consonants that share the same position with respect to the lines on ruled paper, when letters are handwritten on the lines (Figure 1a). These groups are: (1) consonants exceeding the top line (b, d, h, k, l, t); (2) consonants between the lines (c, m, n, r, s, v, w, x, z); and (3) consonants exceeding the bottom line (g, j, p, q, y). This classification of letters is quite intuitive because it is easy to associate a letter with its drawing when using appropriate icons on top of the keys (Figure 1b). When handwritten, the “f” letter exceeds both lines. Performance reasons should be taken into consideration for choosing group 1 or 3 (Figure 1b), e.g. minimal value of KSPC (keystrokes per character) [15].

Vowels should be treated in a different way because all of them belong to group 2 and the sum of their frequencies in English represents more than 26% of all possible letter frequencies. In order to improve performance, it is necessary to distribute the vowels among different keys. The following requirements should be fulfilled in order to make it easier for users to learn the layout: (1) only consecutive alphabetical-ordered vowels can be assigned to a key; and (2) the keyboard must be able to display the five vowels in strict alphabetical order. Figure 2 shows some examples of four, three and two-key keyboards with the vowels printed on the keys. It is possible to propose different layouts for the keyboard families with four, three and two keys. Given a disambiguation algorithm, the next step is to decide which layout is

the most effective by analyzing them, as we have done in the following section.



(a)

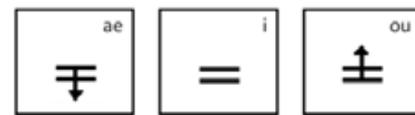


(b)

Figure 1: (a) Grouping of consonants. (b) Keys with icons representing the sets of consonants



(a)



(b)



(c)

Figure 2: (a) Examples of 4 -key (a), 3-key (b), and 2-key (c) CGA keyboards

4. MATERIALS AND METHODS

4.1 Analytical Evaluation

CGA specifications enable the proposal of several layouts for each family of keyboards. Once these layouts are obtained, the following step is to select the best layout. Nevertheless, it depends on several variables that have a direct influence on its performance, mainly the user language, the disambiguation method, the linguistic model or even the physical input device. Although the disambiguation modes WLDM and CLDM, the linguistic model employed and the mathematical model used to evaluate the layout performance are fully described in [17], we introduce them briefly.

4.1.1 Linguistic Model

The linguistic model used in this research is the same as that proposed in [17] and is particularized for each DM. (A) WLDM uses a word-frequency list derived from the British National Corpus and contains the 10,911 most common words in English. The first phase (cell selection phase) uses a 1-gram model obtained from the list and particularized for each character position in a word (e.g. 1st character, 2nd character...). The second phase (disambiguation phase) uses the list to propose words that share the same sequence of selected cells. (B) CLDM uses again a word-frequency list derived from the British National Corpus; nevertheless it contains the 160,258 most common words in English. The linguistic model for the selection phase is the same as before, but now it uses the long list of words. The disambiguation phase uses a linguistic model based on k-grams for the first four characters on each word, and n-grams of fourth order (4-grams) for the remaining characters.

4.1.2 Mathematical Model

The mathematical model is fully described in [17] and it is based on models described by Rosen and Goodenough-Trepagnier [20], Damper [4] and MacKenzie and Tanaka-Ishii [13]. It first estimates the average weighted number of keystrokes per word (n_{weighted}) as shown in Equation 1.

$$n_{\text{weighted}} = \sum_w n_{\text{weighted}}(w) \cdot P(w) = w_S \cdot n_S + w_C \cdot n_C + w_L \cdot n_L \quad (1)$$

In Equation 1, $n_{\text{weighted}}(w)$ is the average weighted number of keystrokes of word w , $P(w)$ is the probability of w ; w_S , w_C , and w_L are the weights for the scan cycles, the clicks and the long presses on the switch, respectively; n_S , n_C and n_L are the average values of the respective number of keystrokes per word.

Text entry speed can be calculated by means of Equation 2, where T is the scan period, and " τ " the time needed to enter a word.

$$\frac{1}{\tau} = \frac{60}{T \cdot n_{\text{weighted}}} \quad (\text{wpm})$$

4.1.3 Procedure

A computer program was written in Microsoft® Excel 2007, more specifically with Visual Basic® for Application (VBA), to evaluate the proposals. This program uses the corpus summary files, and using the previous equations, calculates the average values of the number of scan cycles, the number of switch presses, the number of long presses and finally, the average weighted number of keystrokes.

From this data, it estimates the text entry speed for a scan period of 0.5 seconds. The evaluation of the weighted number of keystrokes uses the reduced word-frequency list with the 10,911 most common words in English instead of several sample text documents. This avoids the bias when using a limited number of text documents.

4.2 Experiment Background

4.2.1 Apparatus: Prototype Description

A prototype application was developed in Visual Basic® to implement a scanning system for testing purposes that implements both disambiguation modes: WLDM and CLDM. It used the CGA3-122 layout. Figure 3 shows the main window, common for the cell selection phase of both DM. A test phrase is shown one at a time, and the user presses the switch when the key with the desired character is highlighted. Next, we present in more detail the operation of both DM in the prototype.

WLDM. The operation of WLDM is composed of two phases: In the cell selection phase (Figure 3), the cursor is initially located on the most probable cell, and if it is not the desired one, the cursor advances to the next most probable cell, using automatic scanning. When the user presses the switch, the input is accepted, generating the code for that cell. All letters in the word except the last one are entered the same way. When introducing the last character, users must keep the switch pressed. This action reports the system that the word is over, and now the second phase (disambiguation phase) begins. Words that share the same code are displayed one after another using inverse scanning. These words are shown on a larger key which replaces the three keys with characters (Figures 4 and 5). When the desired word appears, the user releases the switch and this word is accepted. Notice that the most probable words are shown first, in order to minimize the number of scan cycles.



Figure 3: Main window

CLDM. The operation of CLDM is also composed of two phases: The cell selection phase is identical as in WLDM (Figure 3), except that after clicking the

switch the second phase starts. Now, letters in the cell are displayed again on the larger key³ replacing the three previous keys, and presented one after another using automatic scanning (Figures 6 and 7). The user clicks the switch again when the desired letter appears, and a new cycle begins for the next character in the word. All letters of a word are treated the same way except the last one. Keeping the switch pressed in the first phase when entering last letter tells the system that a space is required after that letter. The scanning mode in the second phase is different because the switch is still pressed. Now inverse scanning is used, and the selection is done simply by releasing the switch.

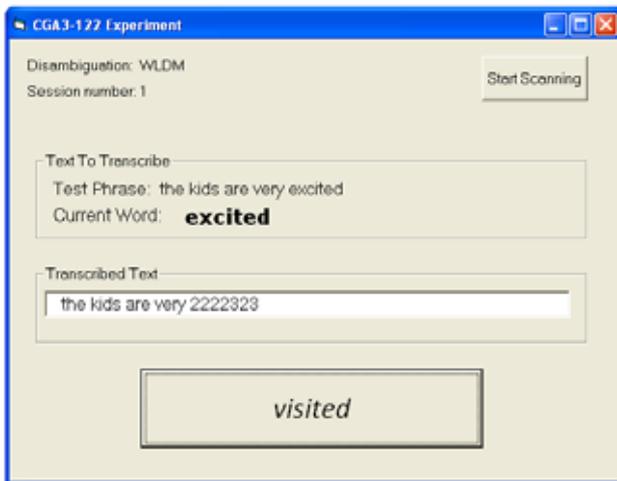


Figure 4: Phase 2 in WLDM

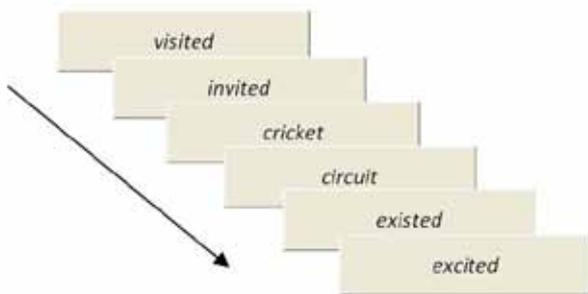


Figure 5: Proposed words in phase 2 of WLDM



Figure 6: Phase 2 in CLDM

³ Notice that the "large key" shows the word being entered and the "proposed" letter.

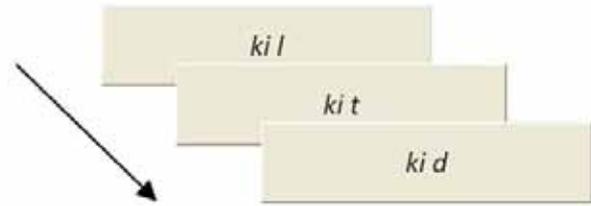


Figure 7: Proposed characters in phase 2 of CLDM

4.2.2 Scan Period and Errors

The scan period can be automatically adjusted while users type the phrases, depending on the number of errors in each phrase. The algorithm is based on [11] and is able to detect and register three types of errors for each phrase:

- Selection error. An erroneous cell, character or word has been selected.
- Timing error. The item has been correctly selected but out of time, i.e., the item has been highlighted twice or more times before pressing the switch.
- Long press error. The switch has not been kept pressed when entering the last character of a word, or a long press has been done when introducing any other character of a word.

The criteria used to adjust the scan period were as follows:

- The scan period increases if the number of errors in any of the previous types of errors is greater than or equal to 3. Specifically the current value is multiplied by 1.05.
- The scan period decreases if the number of errors for all types of errors is less than 3. In particular, the current value is multiplied by 0.95.

After entering each phrase, the information about errors and the new scan period are shown to the user. After pressing the switch, a new phrase is presented for typing.

The prototype displays several short phrases to the user in random order, obtained from a file containing a subset of the 500 phrases proposed by Mackenzie and Soukoreff [12]. Specifically, 133 phrases were used trying to avoid phrases which present some difficulties or unfamiliar words.

After completion of each sentence, the scan period decreases if the previous error limits are not exceeded. Nevertheless, at one point, two participants could reach the same scan period

with different error rates, because of the pass/fail condition for decreasing/increasing the scan period.

To evaluate the performance taking into consideration both the scan period and the error rate, we have defined a figure of merit, which could be defined as follows for each phrase completion:

$$fm_x = accuracy \times \frac{x}{T_{scan}} \times 100$$

In Equation 3, fm_x is the figure of merit; x is the target scan period and is the lowest scan period allowed in the prototype⁴; T_{scan} is the current scan period; and accuracy (Equation 4) is the rate of correct characters in the phrase.

$$accuracy = \frac{N_C - (N_{SE} + N_{TE} + N_{LPE})}{N_C}$$

In Equation 4, N_C is the number of characters in the phrase without considering the blanks; N_{SE} , N_{TE} and N_{LPE} correspond to the number of selection, timing and long press errors, respectively.

The figure of merit increases when the scan period decreases or when the participant decreases the number of errors. Thus, its value will reach 100 when the participant types without errors and achieves the target scan period. This way, if two participants enter a phrase with the same scan period, the participant who makes fewer errors will have a greater figure of merit. Moreover, the figure of merit gives an idea of the improvement achieved by a user during any period of time.

4.3 Experiment

4.3.1 Participants

Six participants recruited from the local university volunteered for the experiment. All the users were non-disabled and they had no previous experience with any kind of assistive technology. There were two females and four males of ages ranging 20 and 25.

4.3.2 Procedure

All participants tested both DM (WLDM and CLDM). To compensate for potential learning effects due to the order in which both DM were tested, participants were split into two groups. Three participants entered text with the WLDM first, followed by CLDM. The other group reversed the order.

The experiment was conducted in a computer classroom with the same physical conditions for all participants. The test for each disambiguation mode took two days, so the total time to conduct the experiment was four days.

⁴The lowest scan period allowed in the prototype was set to 0.35 seconds to prevent users from getting exhausted.

The tasks performed each day were divided into 2 blocks with a break between blocks of one hour. Each block consisted of 3 sessions of 15 minutes with a 5-minute break between sessions of the same block. Therefore, each of the disambiguation modes consisted of 4 blocks or 12 sessions spread over 2 days.

Before the start of session 1, participants performed a training session and entered 4 different phrases, (identical for all participants). The second day, just before the start of the 7th session, participants performed a shorter training session and entered 2 different phrases (again identical for all participants). The instructions were to enter each phrase "as quickly as possible while trying not to make mistakes".

The procedure followed in both disambiguation modes was as follows. Initially the system displays a phrase to be memorized by the participant. Moreover, the word being typed is displayed in the next line using a larger font size (Figure 3). After clicking the switch, the phrase is entered, and registered by the system. When the last word in a phrase is just finished, the systems stops and displays the performance and error data for that phrase in a new window. When the participant closes the window, a new phrase is displayed to be memorized. After the 15 minutes allotted to the session and after completion the phrase, the system shows an end-of-session message and closes the application.

The scan period was set to 1.2 seconds at the beginning of session 1. Successive sessions started with the scan period reached in the previous session.

4.3.3 Design

The factors considered were: Disambiguation mode, Session and Group.

The experiment was treated as 2 x 2 x 12 mixed factorial design. Group is the between-subjects factor (Group A and Group B, with three participants each). The within factors were Disambiguation Mode (WLDM, CLDM), and Session (Session 1 to Session 12). A total of 144 data entries were evaluated (3 participants / group x 2 groups x 2 modes x 12 sessions = 144).

The dependent variables were phrase insertion time and scan period. The phrase insertion time was converted to wpm, using the average word length of the set of phrases, which was 4.3 characters.

5. RESULTS AND DISCUSSION

5.1 Analytical Evaluation

In our research, we evaluated 458 four-key layouts (CGA4-xx), 126 three-key layouts (CGA3-xx) and 60 two-key layouts (CGA2-xx) using both DM. Table 1 presents the performance parameters for the best layout in each family of keyboards. The first column shows the given name for the layouts⁵. The second and third columns show the average weighted number of keystrokes per word (n_{weighted}) for both WLDM and CLDM respectively. The fourth and fifth columns show the estimated text input speed for both DM. The letter-to-key assignments for these layouts can be seen in Table 2.

Table 1: Basic performance parameter for the best CGA layouts in 4, 3 and 2-key families

Code	n_{WLDM}	n_{CLDM}	wpm_{WLDM}	wpm_{CLDM}
CGA4-10	4.03	6.21	16.48	10.70
CGA3-122	3.95	6.13	16.66	10.79
CGA2-9	5.44	6.65	13.81	10.30

Table 2: Letter-to-key assignment for the best CGA layouts in 4, 3 and 2-key families

Code	Keys	Layout
CGA4-10	4	{abdfhkl} {eicmnrsvwxz} {ogjppy} {u}
CGA3-122	3	{agjppy} {eicmnrsvwxz} {oubdfhkl}
CGA2-9	2	{aecmnrsvwxz} {ioubdfhklgjpqy}

As seen in Table 1, CGA3-122 presents a higher performance in both DM, which corresponds to a lower value of n_{weighted} , or that is to say, a higher text input speed in wpm.

5.2 Experiment

The grand mean for entry speed was 9.3 wpm with large differences between disambiguation modes: 12.2 wpm in WLDM (SD=2.7) and 6.3 wpm in CLDM (SD=0.9).

The grand mean for scan period was 0.63 seconds and the means for both disambiguation modes were: 0.53 seconds in WLDM (SD=0.20) and 0.72 seconds in CLDM (SD=0.15).

Figure 8 shows the average speeds achieved in each of the sessions for both disambiguation modes.

The highest speeds recorded for a session were 16.9 wpm in session 12 of WLDM and 8.3 wpm in session 10 of CLDM. The highest speed in WLDM is greater than the speed predicted in the analytical evaluation (16.7 wpm). This is due to the fact that in our calculations, we used a scan period of 0.5 seconds, and this participant reached lower scan periods.

⁵ The general name is CGAx-y, where “x” is the number of keys, and “y” is a consecutive number assigned to this layout in the x-key family.

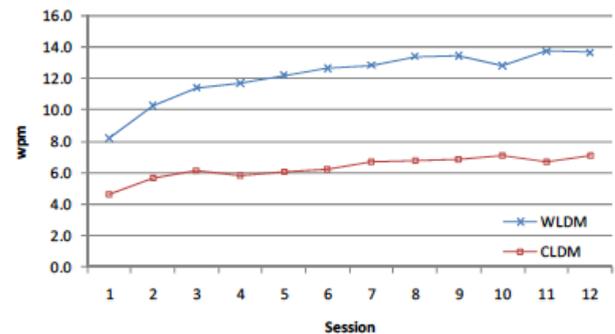


Figure 8: Entry speed (wpm) by disambiguation mode and session

Table 3 shows the percentage of improvement in the text entry speed between session 1 and session 12 for both WLDM and CLDM.

Table 3: Speed improvement with practice

Disamb. Mode	Session 1	Session 12	% improvement
WLDM	8.2	13.7	67.1%
CLDM	4.6	7.1	54.3%

Regression models for both disambiguation modes were obtained to observe the power law of practice [3]. Prediction equations and R2 values are shown in Figure 9. The high R2 values imply that the adjusted learning models provide a good prediction of user behaviour.

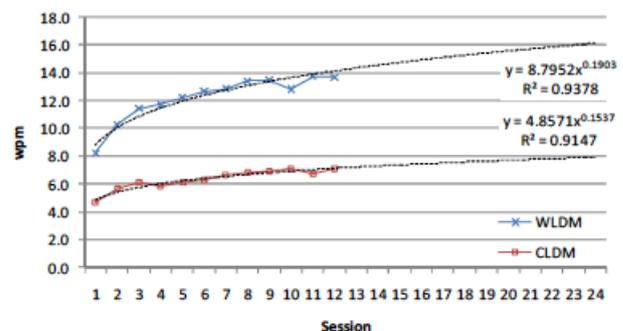


Figure 9: Entry speed learning curves in wpm and extrapolation to Session 24

As discussed previously, the scan period was

gradually adjusted to the user conditions. Figure 10 shows a decrease in the scan period when increasing the number of sessions. Regression models for both disambiguation modes were obtained to estimate scan period values for future sessions. Prediction equations and R2 coefficients are also shown in Figure 10. R2 values remain high.

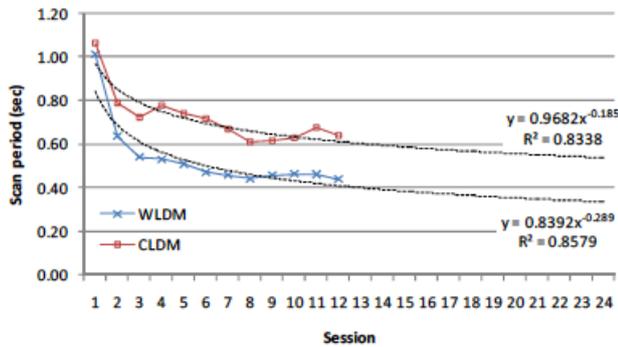


Figure 10: Scan periods and extrapolation to Session 24

Figure 11 shows the figure of merit for a target scan period of 0.35 seconds. As shown, the figure of merit of WLDM always exceeds that of CLDM for all sessions, so WLDM is always more efficient (fewer errors and/or lower scan period).

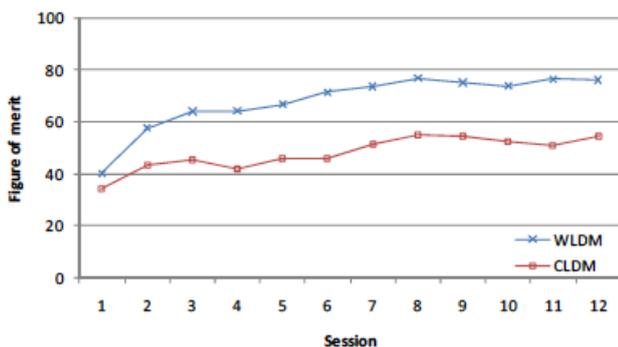


Figure 11: Figure of merit for a target scan period of 0.35 seconds

The main reasons why the figure of merit is less in CLDM are fairly obvious. First, the scan period is larger in CLDM than in WLDM while target scan period remains constant for both disambiguation modes, i.e. 0.35 seconds. Moreover, CLDM requires more keystrokes than WLDM as we estimated previously. With a larger number of keystrokes for entering any given phrase, the number of errors should be larger. As a result, both features cause a decrease in the figure of merit.

Given the results of the experiment, it is possible to draw the conclusions presented below.

CGA3-122 keyboard can achieve speeds of 13.7 and 7.1 wpm for WLDM and CLDM respectively in the twelfth session. These are quite high speeds for a scanning system, especially considering a learning period of only 180 minutes. The top speeds reached

in the experiment were 16.9 and 8.3 wpm in WLDM and CLDM respectively.

WLDM is more efficient than CLDM since:

- It reaches a higher text entry speed from the first session.
- The figure of merit is higher, which means fewer errors and/or a lower scan period.
- The superior percentage of improvement suggests that users' learning process is faster in WLDM.
- The total number of keystrokes is smaller, resulting in fewer errors and less physical effort in WLDM.

With practice, a WLDM user could enter text with a higher speed than the speed evaluated in the theoretical study because, after a few sessions, the scan period has been less than 0.5 seconds.

The differences in the scan period for both WLDM and CLDM suggest that differences in the text entry speed between them will be even greater than those derived from the theoretical study, since in the latter, the scan period was set at 0.5 seconds for both disambiguation modes.

However, WLDM only allows the insertion of the words included in the dictionary. Therefore, another disambiguation mode is necessary to permit a user to enter any word, such as CLDM.

6. PLANNED LINES OF RESEARCH

We are currently developing a prototype scanning system intended for the Spanish language. We are also incorporating basic properties such as editing functions and other sets of characters, mainly numbers and punctuation characters. Our next step is to conduct experimental tests with people with severe motor disabilities in order to evaluate how performance is affected by the nature and degree of a user's disability.

7. ACKNOWLEDGMENTS

This study has been partly funded by Cátedra Telefónica - UPV.

8. REFERENCES

[1] Belatar, M. and Poirier, F. (2008) 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, 209-216.
- [2] Biswas, P. and Robinson, P. (2008) Automatic Evaluation of Assistive Interfaces. In Proceedings of the ACM International Conference on Intelligent User Interfaces, 247-256.
- [3] Card, S.K., Newell, A., and Moran, T.P. (1983) *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- [4] Damper, R.I. (1984) Text composition by the physically disable: A rate prediction model for scanning input. *Appl. Ergon.*, 15, 289-296.
- [5] Darragh, J.J. and Witten, I.H. (1992) *The Reactive Keyboard*. Cambridge University Press, New York.
- [6] Grover, D.L., King, M.T. and Kuschler, C.A. (1998) Reduced keyboard disambiguating computer. Patent No. US5818437, Tegic Communications, Inc., Seattle, WA.
- [7] Harbusch, K. and Kühn, M. (2003) An evaluation study of two-button scanning with ambiguous keyboards. In Proceedings of the 7th European Conference for the Advancement of Assistive Technology, 954-958.
- [8] Harbusch, K. and Kühn, M. (2003) Towards an adaptive communication aid with text input from ambiguous keyboards. In Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics EACL '03, 207-210.
- [9] Kühn, M. and Garbe, J. (2001) Predictive and highly ambiguous typing for a severely speech and motion impaired user. In *Universal Access in Human-Computer Interaction*, C. Stephanidis, Ed. Lawrence Erlbaum, Mahwah, NJ, 933-937.
- [10] Lesh, G.W., Moulton, B.J., and Higginbotham, D.J. (1998) Techniques for augmenting scanning communication. *Augment. Altern. Comm.* 14, 81-101.
- [11] Lesh, G.W., Higginbotham, D.J., and Moulton, B.J. (2000) Techniques for Automatically Updating Scanning Delays. In Proceedings of the RESNA 2000 Annual Conference, 85-87.
- [12] MacKenzie, I.S. and Soukoreff, R.W. (2003) Phrase sets for evaluating text entry techniques. In Proceedings of Extended Abstracts of the ACM Conference on Human Factors in Computing Systems CHI 2003, 754-755.
- [13] MacKenzie, I.S. and Tanaka-Ishii, K. (2007) Text entry using a small number of buttons. In *Text Entry Systems: Mobility, Accessibility, Universality*, I. S. MacKenzie and K. Tanaka-Ishii, Eds. Morgan Kaufmann, 105-121.
- [14] MacKenzie, I.S. (2009) The one-key challenge: Searching for a fast one-key text entry method. In Proceedings of the ACM Conference on Computers and Accessibility, 91-98.
- [15] MacKenzie, I.S. (2002) KSPC (keystrokes per character) as a characteristic of text entry techniques. In Proceedings of the International Symposium on Mobile Human-Computer Interaction, 195-210.
- [16] MacKenzie, I.S., Kober, H., Smith, D., Jones, T., and Skepner, E. (2001) LetterWise: Prefix-based disambiguation for mobile text input. In Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology UIST '01, 111-120.
- [17] Miró-Borrás, J. and Bernabeu-Soler, P. (2009) Text Entry in the E-Commerce Age: Two Proposals for the Severely Handicapped. *J. Theor. Appl. Electron. Commer. Res.* 4, 1, 101-112.
- [18] Miró, J. and Bernabeu, P.A. (2008) Text entry system based on a minimal scan matrix for severely physically handicapped people. In Proceedings of the 11th International Conference on Computers Helping People with Special Needs, 1216-1219.
- [19] Miró-Borrás, J., Bernabeu-Soler, P., Llinares, R., and Igual, J. (2009) Ambiguous Keyboards and Scanning: the Relevance of the Cell Selection Phase. In Proceedings of 12th IFIP Conference on Human-Computer Interaction INTERACT 2009, Part II, 1-4.
- [20] Rosen, M. J. and Goodenough-Trepagnier, C. (1981) Factors affecting communication rate in non-vocal communications systems. In Proceedings of the 4th Annual Conference on Rehabilitation Engineering, 194-196.
- [21] Vanderheiden, G.C. (1988) Overview of the basic selection techniques for augmentative communication: Past and future. In *The Vocally Impaired: Clinical Practice and Research*, L. Berstein, Ed. Grune & Stratton, 5-39.