This text describes much of the testing and development of two bit quartering with the reliable experiments necessary to design and refine it. Though some of it is outdated in the sense that many details have changed since then and last implementations of two bit quartering are faster, the results are still true. As some of it was done with the author as a test-subject, an obvious necessity because results had to be reproducible and highly reliable, and as I have extremely frustrating experiences with scientific journals, minimal effort was spent trying to get it published.

To get a feeling for two bit quartering it is better to read twobit quartering, 2008, see the demo and possibly download etyp.

Joris Verrips, march 2008

Running head: *Active Scanning*

Three Techniques

for Active Scanning

Joris Verrips, MD

Paramaribostraat 138"

1058 VP Amsterdam

j.verrips@planet.nl

Three Techniques for Active Scanning

Keywords: Scanning, Morse code, voice output communication aid, user modelling, input method.

Abstract

Subject Scanning techniques allow speechless people to communicate with electronic switches. With classical passive row-column scanning users wait as first different rows are suggested and then different columns. With active column-row scanning and with active group-wise scanning users move the focus themselves.

Object This study analyses different scanning techniques, models what users have to do to operate them and investigates problems encountered in their design and optimisation.

Method A model is presented of time per keystroke as a function of pause time. Performance was tested with pauses of one second and shorter. Ease of learning was documented with several n=1 experiments. Also, switch counts with several techniques were compared based on relative frequencies of characters in printer's lay.

Results With pause times of one second speeds were 12.3 characters per minute (cpm) for passive scanning and 14.7 to 22.0 cpm for different forms of active scanning. When pause times were adapted to individual needs, highest speeds with a single switch were 22 cpm with passive scanning and 25.1 cpm with active scanning. With two switches highest speeds were above 30 cpm with several techniques. With a small subset of two-switch Morse code to guide a shrinking focus and frequency optimised matrices for group-wise scanning, 30.5 cpm was documented with the screen turned off, a copy task maintained during thirty minutes and an error rate of 3% only.

Conclusion Active scanning techniques are effective alternatives to passive scanning techniques.

Some people with severe motor disabilities can neither write nor type. If they have lost the ability to speak, they may scan the alphabet, aided by an interpreter, to select individual characters and spell words. Hi-tech scanning replaces the interpreter by a machine and may combine an on-screen keyboard with electronic switches. This article analyses scanning techniques and describes learning with healthy test subjects and prolonged copy tasks. It is intended for readers with background knowledge in Augmentative and Alternative Communication; for a simplified account consult Verrips, 2002.

Users of passive scanning techniques wait for the machine to suggest either single items or groups of a fixed size such as a row or a column. When the correct item is highlighted, a click with a switch selects it. Active scanning replaces pauses by clicks and vice versa; users move the focus with clicks and wait to see the machine accept the correct item. Three different forms will concern us: active scanning with a single switch, known as converse scanning, active cr-scanning with two switches, that vaguely resembles a technique known as step-scanning, and active group-wise scanning, that resembles a technique called successive quartering with variable depth but uses Morse code instead of a joystick.

Alternatives to scanning exist. With limited but usable hand function or neck function a laser light can point to a keyboard with light sensitive cells; a mouse or a mouse pen may likewise be used to select on a projected keyboard. Eye tracking systems and Darci Too code with a joystick are fast and appropriate for small patient groups (note 1). Morse code becomes almost automatic with continued practice and does not require a computer (Anson, 1997; King, 2000). However, Morse requires a significant learning effort by potential users and, in low-tech applications, by family members and caregivers as well.

Short review of literature

Vanderheiden (1988), provided an excellent overview of input techniques as direct selection, encoding and different types of time-dependent scanning. Recent overviews of research on scanning and related human factors literature are in Koester and Levine (1994), Demasco (1994), Lesher, Moulton and Higginbotham (1998), as well as Venkatagiri (1999). Also, Damper (1984) is worth reading. The size and content of the scan matrices, the distance travelled by the moving focus, switch counts as related to information theory and learning by potential subjects have all received attention, though certainly there is room for more research.

Two recent studies are of importance. Lesher et al. (1998) have simulated several dynamically changing scan matrices integrated with character prediction and word prediction and have reported savings in the number of switch presses of up to 39.58%. They advocate empirical studies with human subjects to determine "whether these savings can offset the increased cognitive loads associated with the more complex designs". It is almost self-evident that such studies are no easy task.

Venkatagiri (1999) has simulated passive row-column scanning with different matrix layouts and a database of 260 English phrases. He has shown that, with pauses of 1000 milliseconds (ms) and a frequency-optimised scan matrix, passive row-column scanning cannot be faster than 12.53 characters per minute, see Table 1. Many of the experiments described below use the same database that he kindly made available. At the end of this careful study he writes "This underscores the need to employ additional rate enhancement techniques available to AAC consumers ... as well as the need to break new ground in AAC communication technology." It is the belief of the author that to advance the field of AAC, informal experiments with healthy subjects are essential, because they allow easy testing and redesign, and should be discussed in writing before patient studies are in order.

Passive column-row scanning with a single switch

Many articles contain phrases like "The scanning technique involves the sequential presentation of message elements and the user's signal at the proper time to select the desired element" (Ratcliff, 1994, p 67) or "Scanning - any technique where the selection is made based on the moment in time when a user makes a signal" (Vanderheiden, 1988, p 8). To understand the passive scanning process in detail, consider the following example.

To accept item 3,4 (row: 3, column: 4) in Matrix 1 with passive column-row scanning (cr-scanning; note 2) one must start with a click (note 3), wait three pauses for the focus to move from 1,1 to 1,4, click again, wait two pauses to see the focus move from 1,4 to 3,4 then click a third time to accept. The total is three clicks and five pauses (note 4); we may assume five decisions as well. The pause is the amount of time allowed to click before the focus moves on and therefore pauses take more time than clicks. The focus moves by itself and users change its direction and accept one of its suggestions with clicks; therefore, users have a rather passive role.

(* Matrices 1, 2 about here *)

Analysis

There are several reasons why waiting for a moving focus is error-prone and cognitively demanding. First, it requires a user's undivided attention to select the right cell, and at a pre-set pace. Second, it seems few people appreciate suggestions from a machine. On the contrary, if a machine is used as a communication tool, an excellent user-interface should minimise the need for users to look at the screen. Third, and most important, the law of Hick-Hyman in psychology (Hick, 1952) says that choice times depend on the number of options to be chosen from and start at about 1 second for a choice out of two, see Appendix 1. This implies that users of a passive scan matrix spend most of their time making decisions. Though it looks simple, passive scanning is hard work and can be very frustrating. Trained users will anticipate on the distance that the focus has to travel. They will wait a variable pause and click, to limit the number of decisions they have to make. Active scanning allows them to issue a variable number of clicks and wait, see below.

Active cr-scanning with a single switch

Active cr-scanning is known in the literature as converse scanning. It is based on the observation that clicks are faster than pauses. Users click once to start and three times to move the focus to 1,4, wait one pause to see the appearance of the focus change, click twice to move the focus to 3,4 then wait one last pause to see the machine accept after six clicks and two pauses, see Matrix 3 to 5. Users actively move the focus but it changes direction and accepts by itself, when it times out on a pause (note 5).

(* Matrix 3, 4, 5 here *)

Active cr-scanning with two switches

With free access to two switches, users may reserve one for rows and the other for columns. A pause accepts and this resembles step-scanning as in Vanderheiden (1988). Different paths are possible to move the focus from 1,1 to 3,4; users then wait only one pause to accept field 3,4 by the machine. For this reason active cr-scanning with two switches is somewhat faster than active cr-scanning with a single switch in the case of users who can access two switches.

Active group-wise scanning with two switches

Morse code allows a user to choose from groups of diminishing size. With successive quartering, users need four codes; . .. - and -- as shown in Matrix 6 to 9. It is based on a simple mathematical fact, that one may easily divide a square into four equal and smaller squares, see Figure 1. Three choices from four codes may select 4^3=4x4x4=64 keys that are accepted by the machine once the focus is small enough. This technique lies in between direct access of an item, as in the case of Morse, Morse encoding of isolated icons, or pointing as with a mouse, and sequential access of items as in passive cr-scanning. A small subset of Morse is learned rapidly and functions as an intermediate code. Two more codes have length two: -. corrects if the wrong code was chosen and .- acts as a shortcut (note 6) that is usually bound to space, the most frequently used character. This technique resembles successive quartering with a joystick as described in Vanderheiden (1988) and, though it is here considered as a form of scanning, is not time dependent (note 7). It may be considered as a special case of encoding and appears to be new.

(* Figure 1, Matrix 6 to 9 here *)

Test of two active scanning techniques

Two active scanning techniques were tested by the author to compare with passive scanning. The same data were used as in the study by Venkatagiri (1999) and the same pause settings of one second. In the hope to facilitate learning, layouts of the scan matrices were different, matrix 3 to 6 instead of matrix 2. Rates were measured after much practice, including tests during development, twelve hours for group-wise scanning and four hours for active cr-scanning with a single switch.

Procedure

Four consecutive groups of eight lines (4x8=32 lines with a total of 1027 characters) were new to the author. Each group was entered twice, either with active scanning with a single switch (A) or with group-wise scanning (B): AB AB BA BA. Synthetic speech was on, keys were not spoken and in condition A clicks were made audible and varied in pitch with the place of the focus. In condition A a single switch pressed for a period longer than 500 ms was interpreted as several clicks and was counted as several clicks too. This function resembles so-called two-speed scanning and was rarely used.

(* Tables 1 and 2 here *)

Results and interpretation

Given that 12.53 cpm (chars per minute) is the highest speed obtainable with passive cr-scanning and pauses of 1000 ms these speeds are higher ($p<0.01$ with t-test, one-sided) for both A and B, see Table 2 and note the low measures of variance. The small difference between A and B is not important because pause times were not adapted individually; see Appendix 2 for models of scanning speed as a function of pause times.

Optimised Active Scanning

To speed up cr-scanning one may consider a frequency-based layout as in Matrix 2. In Matrix 10, a first row has been added with empty fields that the focus cannot accept. This allows a variable rest between column scanning and row scanning. Keys such as arrows and backspace are often repeated. If the focus restarts there, they may be repeated with a single click. Once the need for repetition is over the focus may be pushed to 1,1. If special switches make a difference awaits further research, see note 10.

To improve group-wise scanning we will consider different matrix layouts, introduction of more Morse codes, an extended role for the shortcut and pause reduction. If the number of choices for frequent characters like a, e, i, n, o, r, s or t is reduced they need more space as in Matrix 11 where . -- selects e and . .. selects a. This is known as variable depth successive quartering, see Vanderheiden (1988) and note 11. Other Morse codes may bypass the matrix such as ..--=SP, ----=BS, .----=1, ..---=2 as illustrated in Figure 2. The shortcut .- is shifted by a first choice to SP (Space) if the first quarter is chosen, to . (Period) if the second quarter is chosen, to Return with the third quarter or to , (Comma) with the fourth quarter; .. .- will select Return. Arrows and other cursor movement keys set the shortcut to allow easy repetition, much like the feature described above to optimise cr-scanning, and other subtle adaptations are possible, see note 12.

(* Figure 2 and Matrix 10-12 here *)

The last optimisation for group-wise scanning is rather subtle. About 60% of used codes have length two. To speed up, the software may NOT pause after two switches. Codes . or - are followed by a pause but .. .- -- and -. are not; to enter longer bypassing codes becomes impossible. With scan matrix 6 we estimate 3 c(odes)+1.2 p(auses) per character as 40%*3=1.2; pause reduction leads to 2.25 c+1.45 p, according to Appendix 2.

Questions

There is no lack of reasonable questions about optimised active scanning techniques. One may wonder how fast they are, how learnable and how trustworthy in the long run, in what contexts they can be applied, and more. A coordinated research effort is needed to answer these questions with dependent users, with due care of course to ethical aspects. For the moment, improvised laboratory studies and a few studies done by others are all the information that is available. For details about experiments that test optimisations and document speeds above thirty cpm see the next section. Rapid group-wise scanning has been replicated with several other test subjects, see the last section.

More tests by the author

During development, many experiments were done with the author as a test subject. If with design is meant the process of gradually adapting and improving existing techniques that is typically repetitive and leads to increased understanding of the problems posed by these techniques, these experiments informed design because they allowed to compare different scanning techniques. They should not be considered as formal evaluation of a communication aid according to some pre-defined methodology but they certainly do test hypotheses and models as described in the appendices and might be reproduced by others.

To compare scanning with a single switch and group-wise scanning with two switches at low pause times four times six phrases were copied (a total of 901 characters) with an ABBA ABBA-design. See Table 3. For a comparison of active cr-scanning with two switches and with different pause times, see Table 4. The experiments reported in Table 5 compared C= passive scanning with a single switch and pauses of 1000 ms, D= passive scanning with a single switch and pauses of 480 ms, E= active cr-scanning with a single switch and F= active cr-scanning with two switches. This time the same phrase 'How much does this cost?' was copied with CCCC DDDD EEEE FFFF, a total of sixty-four times; four times four for every condition of a phrase that was read only once.

(* Tables 3-5 here *)

To document the effect of an empty first row on error production this same phrase was repeated with pauses of 400 ms and passive single switch scanning, G then H in Table 6. This rapidly became impossible; clearly, these pauses are too short for the author with a single switch. As expected, the number of errors in condition H was much higher than in condition G, suggesting that an empty first row with variable rest is comfortable. To document speeds with pause reduction the same phrase was scanned with pauses of 400 ms (I) and 1000 ms (J) and after some more training with Matrix 14. See Table 7 for IIII JJJJ; even with pauses of 1000 ms, optimised group-wise scanning allows reasonable speeds.

(* Tables 6, 7 here *)

Two more hours were trained with Matrix 14, screen turned off, speech turned on, pauses of 280 ms and the same copy task as used before. Rates in the four last sessions were 30.55 cpm (characters per minute) (SD±0.7) with 31 errors in 41 phrases and 1108 characters (3%), 2364 codes (2.1 per character) that consisted of 3406 switches or 3.07 per character (note 13). After continued training and phrases repeated twice in short sessions of one or two minutes, values up to 40 characters per minute have been recorded, including some inevitable errors. Those results seem close to the upper limit of scanning effectiveness and approach rates achieved with Morse code and two switches.

I trained some more with the same phrase as in Table 5, Matrix 12, pauses of 320 ms and single switch group-wise scanning. Speed was 23.43 cpm, SD=0.45, one error occurred in four phrases (1%) and 140 switches were needed for 25 characters, 5.6 per char. See Table 8 for an overview of experiments performed with the author as a test subject.

(* Table 8 here *)

If we concentrate on pause times of one second and put data from the model in appendix 2 in square brackets, rate with passive scanning was 12.3 [12], rates with different forms of active scanning were 14.73 [15], 21.77 [20], 15.63 [15] and 22.01 [27]. See Tables 2, 4, 5 and 7 and note that these rates were not measured with the same texts and that there are evident methodological problems when the author functions as his own test subject. Ideally, experiments would be repeated by independent research with members of relevant patient groups. If the author is considered as a possible patient and the other data are taken into account the following applies. Optimised group-wise scanning with two switches is superior to active cr-scanning with two switches in both speed, attention required and switches pressed per keystroke. In addition, it is feasible with eyes closed or turned elsewhere. With a single switch, active scanning is preferable to passive scanning. When Morse code is an option teaching active scanning first ensures rapid results.

Ease of learning

Active scanning techniques were often demonstrated to obtain reactions and improve screen presentation and ease of learning. Many reactions did eventually lead to specific details, like changing only foreground colours. This section concentrates on the experiences of five different test subjects who learned the system. They do not represent the outcome of a carefully controlled evaluation effort in a well-equipped laboratory but rather anecdotes and figures gathered from precise measurements over a considerable period, often in informal contexts. All test subjects were included but they were not selected to represent a specified patient group; neither were different systems benchmarked against each other by independent research.

Maartje is a friendly lady of age 77 who never had anything to do with computers and helped in return for some instruction. On the first afternoon, we studied the keyboard, discussed computers in general and tried the speaking editor that encapsulates the scanning system. Maartje used cr-scanning with a single switch for 10 minutes and understood it almost immediately. It took a second afternoon and much talk and explanation before she wrote her name with it. She had trouble to wait for the machine to accept characters and needed long pauses because she felt uncertain. Much time was spent to look for characters on the display as in Matrix 3. After a total of four hours hands-on experience and about four hours of talk and tea she achieved eight characters per minute using active scanning with a single switch and pauses of 1400 ms. Her comments inspired the active cr-scanning with two switches and an empty first row. She did not try group-wise scanning.

Three other participants were Leonie, a bright schoolgirl aged 17 with some experience with word processors who also received some instruction in return for her cooperation, Carolien, a paid student of 22, and Arnoud, my son of 13. They needed about 25 minutes of explanation and try-out to understand the speaking editor and the group-wise scanning. Their comments on ease of learning are in Table 9. All test subjects were asked to practice a bit.

Leonie did spend 80 minutes repeatedly copying a song with pauses of 320 ms, Matrix 14 and pause reduction. She then scanned at 18.6 cpm. Carolien practised for 85 minutes copying the same text as used above with optimised group-wise scanning, Matrix 13 and shifted shortcut. Her pause time was adapted to her preference at 320 ms and she then copied 706 characters in 28 minutes with 25 corrected errors (3.5%), 25.06 cpm after a total of 115 minutes hands-on practice, see Table 10. At the time she could scan with her eyes closed and expected further improvement. As she did not put in the same effort with active cr-scanning a direct comparison is impossible.

(* Tables 9 to 10 and Matrix 15 here *)

Arnoud repeatedly copied a Dutch poem he knows by heart and achieved 28.53 cpm after at least two hours of very hard working (rest periods not included) pauses of 320 ms and Matrix 14. He also tried low-tech active group-wise scanning, tapping codes on the author's shoulder and used . Left, .. LeftLeft, - Right and -- RightRight. Luca, a friend of Arnoud aged 12 years and much interested in computers, first looked how Arnoud scanned. He then concentrated on the scanning and achieved 20 cpm with a short Dutch phrase, Matrix 15, pauses of 400 ms and training of only 10 minutes. This very rapid learning cannot be expected with members of relevant patient groups.

Conclusions

1. Active cr-scanning with two switches is faster than active cr-scanning with a single switch.

2. Two-switch Morse as intermediate code allows active group-wise scanning as illustrated in Matrix 6 to 9.

3. Models of scanning show that if pauses are longer than clicks, all forms of active scanning are faster than passive cr-scanning.

4. Scan matrices with an empty row or an empty column allow a variable rest of the moving focus during active cr-scanning.

5. Active group-wise scanning can be optimised with a shifted shortcut, with more space for frequent characters, by the introduction of bypassing Morse codes and by pause reduction.

6. Optimised active group-wise scanning is superior to active cr-scanning in the sense that it requires fewer switch activations and allows scanning with eyes closed.

7. It requires several hours of training to achieve high rates with active scanning techniques.

8. Testing scanning techniques with healthy test-subjects allows to gather reliable and reproducible measurements, that are a necessity to advance the field of AAC. (added in march 2008).

References

Anson, D. (1997). Alternative computer access: a guide to selection. Philadelphia: F.A. Davis.

Card, S. & Moran, T.P. & Newell, A. (1983). *The Psychology of Human Computer Interaction.* Lawrence Erlbaum. New York.

Damper, R.I. (1984). Text composition by the physically disabled: A rate prediction model for scanning input. *Applied Ergonomics*, 15, 4, 289-296.

Demasco, P. (1994). Human factors considerations in the design of language interfaces in AAC. *Assistive Technology,* 6:10-25.

Hick, W.E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology, 4,* 11-16.

Horn, E.M. & Jones, H.A. Comparison of two selection techniques used in augmentative and alternative communication. (1996). *AAC, Augmentative and Alternative Communication, 12, 1996,* 23-31.

Horstmann Koester, H. & Levine, S. (1994). Learning and performance of able-bodied individuals using scanning systems with and without word prediction. *Assistive Technology, 6;42:42-53.*

Horstmann Koester & H.M., & Levine, S.P. (1997). Keystroke-Level models for user performance. *AAC, Augmentative and Alternative Communication, 13,* 239-257.

Huffman, D. (1952). A method for the construction of minimum redundancy codes. *Proceedings IRE, 40,* pp. 1098-1101*, Sept. 1952.*

Hunnicutt, S. & Carlberger, J. (2001). Improving word prediction using Markov models and heuristic methods. *AAC, Augmentative and Alternative Communication. 17 December 2001,* 255-264.

King, T. (2000). Modern Morse code in rehabilitation and education. New applications in assistive technology. Allyn and Bacon.

Lesher, G.W. & Moulton B.J. & Higginbotham D.J. (1998).

Techniques for augmenting scanning communication. AAC, Augmentative and Alternative Communication. 14, June 1998, 81-101.

Mizuko, M. & Reichie, J. & Ratcliff, A. & Esser, J. (1994). Effects of selection techniques and array sizes on short-term visual memory. *AAC, Augmentative and Alternative Communication 10,* 237-244.

Preece, J. & Rogers, Y. & Sharp, H. & Benyon, D. & Holland, S. & Carey, T. (1994). *Human-Computer Interaction.* Addison Wesley.

Ratcliff, A. (1994). Comparison of relative demands implicated in direct selection and scanning: Considerations from normal Children. *AAC, Augmentative and Alternative Communication 10,* 67-74.

Stower C. (1817). 'The Printer's Manual: an abridgement of Stower's Grammar.' Boston:Crocker. Reprinted (1981) New York Garland.

Tanenbaum, A.S. (1984). Structured computer organisation. Prentice Hall Second Edition 1984.

Vanderheiden, G.C. (1988). Overview of the basic selection techniques for augmentative communication. Past and future. In L.E. Bernstein (Ed.), The vocally impaired: Clinical practice and research (pp. 5-39). Philadelphia: Grune and Stratton.

Venkatagiri, H.S. (1999). Efficient keyboard layouts for Sequential Access. *AAC, Augmentative and Alternative Communication 15, June 1999,* 126-134.

Verrips, J. (2002). Letter based techniques. http://home.planet.nl/~verphtyp/lowtech.htm .

Verrips, J. (2000). Test of a communication aid with stored material. *International Journal of Rehabilitation Research, June 2000. 23,* 139-144.

Verrips, J. (1992). Branching selection of suggestions. *Interacting with Computers, 4,* 68-82.

Notes

Note 1 Several commercial websites offer information about eye-track technology: www.eyetechds.com, www.metrovision.fr or www.prentrom.com. For light-sensitive keyboards, consult www.kompagne.nl. Info on Morse is on www.handiham.org, on Morse and Darci Too on www.westest.com. Sensational speeds have been reported with Dasher, that combines eye-track technology, mouse or joystick and both word prediction and letter prediction, and that gives more space to more frequent words: www.inference.phy.cam.ac.uk/dasher/. Trustworthy evaluation studies are scarce, especially with handicapped subjects and with precise data on error rates, as seems inevitable in AAC. Experts from Metrovision treated 14 patients with eye-track technology in Europe and warned me that some contact lenses, dry eye syndrome of brain stem ischaemia, the presence of minor nystagmus, spontaneous head movements, age over 60 and all cognitive problems complicate eye-track technology with lock-in syndrome (B. Vermandel, personal information). In addition, up to 40 hours of practice would be needed to achieve fluency with very variable rates and at a relatively high price. Therefore, one may still research switch-based techniques and hope to see it applied.

Note 2 In the rest of this paper the term and technique of cr-scanning (column-row scanning) is employed instead of the habitual rc-scanning (row-column scanning). Cr-scanning fits better with Matrix 3 that seems easy to learn but is equivalent to rc-scanning in number of switches.

Note 3 With a click is meant a user who activates a switch.

Note 4 We assume no pause to start and one pause to accept field 1,1 with a total of three clicks.

Note 5 The focus changes state and appearance: click-will-start-me to click-shifts-my-column to click-shifts-my-row. It then accepts (e.g. reads) an item and returns to the centre and to its first state as in Matrix 3. When a user holds the switch, the focus continues to move, slowly, to allow planning and coordination. It may also issue sounds when it moves. To compare active and passive scanning consider pauses are ____ and clicks __; __ __ __ __ ____ with active scanning must be faster than the equivalent ____ ____ ____ ____ __ with passive scanning. Active cr-scanning is reminiscent of tap-code once used by prisoners of war who shared a small alphabet matrix and knocked two numbers per character to communicate. How tiring it is to access switches varies from one patient to another.

Note 6 Alt-X is an example of an often-used shortcut.

Note 7 Active group-wise scanning should perhaps be called active two-bit quartering as passive (and active) one-bit quartering is possible too. Items might also consist of icons, pictograms, names, commands as well as keys; codes might be spoken or entered in other ways as by joystick, a Morse key or a single switch. King (2000) gives examples of Morse used to select icons as if from a numbered list, among much else. Anson (1997) offers a recent and interesting overview of alternate access from a mostly technical point of view.

Note 8 Note that several user-processes were ignored, such as deciding which word to spell and speak, attention to follow a focus and spot items in a matrix or small pauses between one character and the next. The book by Card, Moran & Newell (1983) is a good starting point on user modelling and contains a theory-based approach to the design of user-interfaces. It also contains ample statistics and a test that showed motivated subjects who outperformed every keystroke-level model by continued training. Though unpractical in every-day software engineering, this theory was influential in this project, especially Appendices 1-3. There is more to Human Computer Interaction than user modelling, see Preece et al. (1994) for an excellent introduction.

Note 9 Users who intend to try scanning combined with stored text should realise that to actually communicate with a machine is rarely easy. In earlier research, twenty hours were spent to learn to communicate with two speaking computers and word prediction in combined monologues (Verrips, 2000). Pre-stored text was repeated, but less than expected, and word-prediction was used for less than ten percent of the total length. The rate enhancement of word prediction is easily counterbalanced by an increase in cognitive load (Koester & Levine, 1997). Experienced telegraphers have reported rates of Morse up to 40 words per minute. It may be wiser to train Morse code than to combine it with text prediction.

Note 10 To reduce the number of clicks special micro switches might be tried with auto-repeat. These reliably issue several clicks when pressed for a longer period of time and react on shorter time intervals than the keyboard. Such switches have been used for high speed Morse code and just might speed up active cr-scanning either with a single switch or with two switches. Other technical research ideas are to couple to electromyography sensors, to couple to switches that offer tactile feedback (for cr-scanning) and to read Morse code from a single switch. Whether a few hours of training would allow satisfactory low-tech group-wise scanning and how it would compare with tap-code is an example of a non-technical research idea.

Note 11 More space for frequent characters resembles Huffman-code that assigns short codes to frequent symbols (Huffman, 1952, Tanenbaum, 1984) as in the saying that "common things are common". It can be implemented in many different ways that influence ease of learning. The (sum of) frequencies determine the space allotted to (a group of) characters. Less frequent characters are still available on a second layer of the scan matrix accessed by -. in Matrix 9.

Note 12 If combined with word prediction the software might set the shortcut to '1' or to the most recent word in the list before the first choice (as in Verrips, 2000; Verrips, 1992). Some forms of syntax-sensitive word prediction seem to require few suggestions on screen (as in Hunnicutt and Carlberger, 2001) and could combine with shifted shortcut as well. Numerals 1 to 0 may also be shifted as when - .---- means 11 or -- .---- means 21 to quote with a margin menu. An unexplored possibility is to let the shortcut be based on letter prediction instead of being bound to space. This could even combine word prediction and letter prediction, though of course at the price of more learning and more complicated user processes.

Note 13 Though high, these rates are lower than those reported by Koester and Levine (1994) in a different experimental set-up with passive scanning. Between phrases rest periods of at least 20 seconds existed and phrases were read before the clock started. They used very motivated graduate students with average pauses of 300 ms only (Horstmann Koester, personal communication, 2002) and found text generation rates that increased from 20.4 to 37.4 cpm during continued practice with the fastest user at 42.3 cpm and even higher with word prediction. The author achieved only about 22 cpm with classical passive scanning and a repeated single phrase, even for short time periods, and with variable error rates.

	Col 1	Col 2	Col 3	Col 4	Col 5
Row 1 ®	0	1	2	3	4
Row 2 ®	1	2	3	4	5
Row 3 ®	2	3	4	5	6
Row 4 ®	3	4	5	6	7
Row 5 ®	4	5	6	7	8

Matrix 1 Scan matrix with the number of pauses needed to select items by classical passive row-column scanning. Scanning usually starts at the left upper angle in field 1,1 with 0 pauses. The focus now is in field 3,4: row three and column four, and took five pauses to get there. A triangular matrix with empty rightmost lower corner requires less pauses and diminishing column size, but would not make a large difference in the total number of clicks needed.

SP	T	I	H	F	B
E	O	R	C	P	Z
A	S	U	G	Q	Del
N	D	Y	J	,	0
L	W	X	.	1	2
M	K	?	3	4	5
V	'	6	7	8	9
Ret					

Matrix 2   A 43-key keyboard arrangement based on letter-frequencies and used by Venkatagiri (1999) for simulated rc-scanning. SP=space, Ret=Return, Del=Delete. The frequencies of SP, T, E, A, O, I taken together often exceed 50%.

SP	.	a	e	i	m	r	w
Ret	,	b	f	j	n	s	x
1	6	c	g	k	o	t	y
2	7	d	h	l	p	u	z
			■				
3	8	F1	F3	F7	q	v	'
4	9	F2	F4	F8	-	Hom	End
5	0	Esc	F5	F9		¬	Pgu
BS	Dwl		F6	F10	¯	®	Pgd

Matrix 3   Matrix intended for active cr-scanning. The focus now rests in the centre, a click will move it to SP.   selects a second matrix much like Matrix 9. Dwl=delete word to left. Numerals allow to combine with word prediction.

SP	.	**<u>a</u>**	e	i	m	r	w
Ret	,	b	f	j	n	s	x
1	6	c	g	k	o	t	y
2	7	d	h	l	p	u	z
3	8	F1	F3	F7	q	v	'
4	9	F2	F4	F8	-	Hom	End
5	0	Esc	F5	F9		¬	Pgu
BS		Dwl	F6	F10	¯	®	Pgd

Matrix 4 Same as Matrix 3 after three clicks. The focus rests on 'a' and if no more clicks ensue, a pause will pass and we may look at Matrix 5.

<table>
<tr>
<td>SP . |a| e i m r w
Ret , b f j n s x
1 6 c g k o t y
2 7 d h l p u z

3 8 F1 F3 F7 q v '
4 9 F2 F4 F8 - Hom End
5 0 Esc F5 F9 ¬ Pgu
BS Dwl F6 F10 ¯ ® Pgd</td>
<td>1, 8 ® to 1,1</td>
</tr>
<tr>
<td>¯ ¯
from 8,2 to 1,1 from 8,8 to 1,7</td>
<td></td>
</tr>
</table>

Matrix 5 Matrix 3 after three clicks and a pause. The focus changed state (and colour). One more pause and it selects 'a', we will then see Matrix 3 again. With more clicks, the focus will rest on 'b', 'c', 'd' etcetera. Behaviour of the focus at the borders is indicated with arrows. If an arrow key is selected we will see Matrix 3 but the focus then jumps to the arrow with a click and jumps to SP with a second click.

. ——————————.- SP—————————— -

| a b e f | k l o p |

| c d g h | m n q r |

| SP . i j | s t w x |

| Ret , - Esc | u v y z |

|————————————|————————————|

| 1 2 5 6 | F1 F2 F5 F6 |

| 3 4 7 8 | F3 F4 F7 F8 |

| Hom End 9 0 | Pgu F9 F10 |

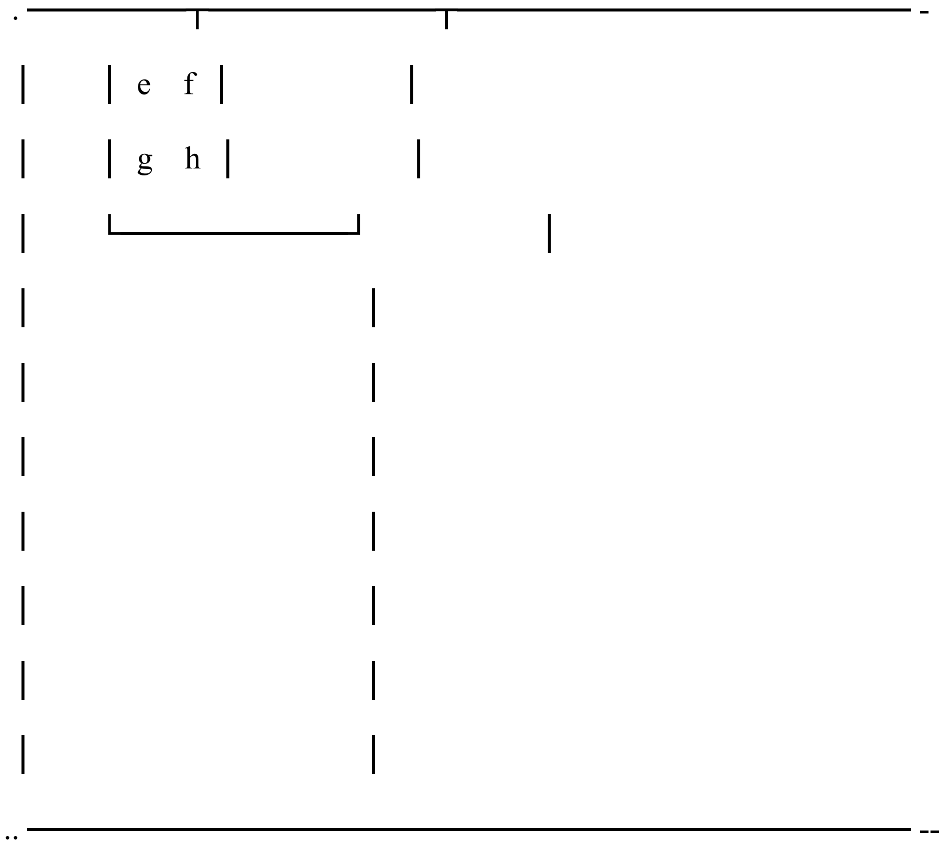
| Dwl BS Del ' | Pgd ¯ ¬ ® |

.. ——————————————————————— --

Matrix 6 Scan matrix intended for group-wise scanning. Codes . - .. and -- are shown in the corners of the quarters they select. . selects the upper leftmost quarter, see Matrix 7. The meaning of shortcut .- is now SP and is displayed on top. -. leads to Matrix 9.

```
. ─────────────────────.- SP ─────────────────────── -
| a   b   e   f  |          |
| c   d   g   h  |          |
| SP  .   i   j  |          |
| Ret ,   -  Esc |          |
├──────────────────────────┘          |
|                   |
|                   |
|                   |
|                   |
.. ──────────────────────────────────────────────── --
```

Matrix 7   The selected quarter is shown, often with a different background colour and/or foreground colour. If - is chosen the upper rightmost quarter will be selected with characters e f g h. -. now corrects and leads to Matrix 6 again.



Matrix 8 . will now select $e - f$, .. g and –– will select h. -.

corrects and will return to Matrix 6.

```
. ------------------------ .- BS ---------------------------- -

| A  B  E  F | K  L  O  P |

| C  D  G  H | M  N  Q  R |

|   è  I  J | S  T  W  X |

| à  é  ë  ó | U  V  Y  Z |

|-----------------------------|--------------------------------|

| [  ]  +  ~ | ï  -  "  ` |

| (  )  *  ^ | &  @  :  ; |

| {  }  _  | | =  %  Dwr  Tab |

| <  >  \  / | #  $  !  ? |

.. ------------------------------|------------------------------ --
```

Matrix 9 Second scan matrix with less frequent characters, usually placed underneath the normal scan matrix and therefore invisible. In Matrix 6 -. displays Matrix 9. Dwr = Delete word to right. The shortcut .- now selects BS.

SP	t	i	h	f	b	Ret	BS
e	o	r	c	p	z	Dwl	
a	s	u	g	q	.	Esc	
n	d	y	j	,	0	Hom	¯
l	w	x	!	1	2	End	®
m	k	?	3	4	5	Pgu	¬
v	'	6	7	8	9	Pdn	-

Matrix 10 Matrix 2 adapted for active cr-scanning with an empty first row to reduce errors while shifting from a horizontally to a vertically moving focus. Pauses only accept if the focus is on a non-empty field, therefore users can rest while the focus waits for them. The focus now starts in field 1,1 and not in the middle. Fields in the two rightmost columns were defined to combine with an editor. Alternatively, edit keys and numerals might be placed on a second layer accessed by in a field close to 1,1.

. -

b	c	g	h	l	m	r	u
d	f	j	k	p	q	v	w
a	a	e	e	n	n	x	y
a	a	e	e	n	n	z	'
i	i	o	o	s	s	t	t
i	i	o	o	s	s	t	t
Hom	End	SP	.	Pgu		-	Esc
Dwl	BS	Ret	,	Pgd	¯	¬	®

.. --

Matrix 11 More space for frequent characters and group-wise scanning leads to fewer codes to select frequent characters; . .. will select a, . – will select e. Other characters are on a second and possibly a third layer, accessed by -. or otherwise.

```
.  ────────────────┬──────────────────  :

| b  c  g  h | l  m  u  v |
| d  f  j  k | p  q  w  x |
| SP  .  Hom End | Pgu  -  y  z |
| Ret  ,  Dwl  BS | Pgd  Esc  !  ' |
├────────────────┼──────────────────┤
| a  a  e  e | n  n  r  r |
| a  a  e  e | n  n  r  r |
| i  i  o  o | s  s  t  t |
| i  i  o  o | s  s  t  t |

:.  ──────────  :.:=SP  ──────────  ::
```

Matrix 12 Optimisation without arrow-keys. Here group-wise scanning is accessed by a single switch : means two clicks and :: four clicks. To select ‘t’ one needs eight clicks, SP requires five clicks. Because the sum of frequencies of characters b to BS (in the first quadrant) is usually much lower than of n, r, s and t in the fourth quadrant, or even of SP, this optimisation can still be improved, possibly at the prize of increased learning-effort.

b	c	g	h	l	m	u	v
d	f	j	k	p	q	w	x
a	a	e	e	n	n	r	r
a	a	e	e	n	n	r	r
i	i	o	o	s	s	t	t
i	i	o	o	s	s	t	t
SP	.	Hom	End	Pgu	-	y	z
Ret	,	Dwl	BS	Pgd	Esc	?	'

Matrix 13 Matrix used with pause reduction: all frequent characters contain one pause, agreeable to beginning users.

```
. ────────────┬──────────────╥──────────────┬──────────── - -
│ b  c │ g  h ║ l  m │ v  w │
│  Hom  │  End ║ BS   │  Del │
│ d  f │ j  k ║ p  q │ x  z │
├──────── SP ────────╫──────── . ────────┤
│ a  a │ e  e ║ n  n │ r  r │
│      │      ║      │      │
│ a  a │ e  e ║ n  n │ r  r │
╞═══════════╪═══════.-=SP═══════╪═══════════╡
│ i  i │ o  o ║ s  s │ t  t │
│      │      ║      │      │
│ i  i │ o  o ║ s  s │ t  t │
├──────── Ret ────────╫──────── , ────────┤
│ u  u │ y  y ║ Ret  BS │   ¯  ½
│      │      ║   '   │  ?  ½
│ u  u │ y  y ║ ,   . │ ¬ ® ½
.. ────────────┴──────────────╨──────────────┴──────────── --
```

Matrix 14 To optimise Matrix 13 u and y need more space, due to their relative frequencies. The meanings of the shortcut .- are displayed in the centres of the quarters that shift it. The reader may wish to verify that - . .- = BS; .. .- = Ret and that - -- selects r with three switches and two pauses. With pause reduction the code length of r is only four switches-or-pauses.

```
1  |
2  | Hello
3  | How are you.
| I like talking @
|

   . ──────────┬──── -. Corrects ────┬────────── - -
   | a  a | e  e || b  c | g  h |
   |      |      || Hom  | End  |
   | a  a | e  e || d  f | j  k |
   ├────── SP ──────╫──────── . ────────┤
   | i  i | o  o || l  m | v  w |
   |      |      ||  '   |  ;   |
   | i  i | o  o || p  q | x  z |
   ╞═══════════╪════════.-=BS═══════╪═══════════╡
   | n  n | r  r || u  u | y  y |
   |      |      ||      |      |
   | n  n | r  r || u  u | y  y |
   ├────── Ret ──────╫──────── , ────────┤
   | s  s | t  t || SP  . | BS   |
   |      |      ||  !   |  ?   |
   | s  s | t  t || Ret , | Del ‾ |
   .. ──────────┴──────────╨──────────┴────────── --
```

Matrix 15 Some subjects prefer this layout to Matrix 14 and find it easier to learn. The shortcut in the middle is set to BS and the meaning of -. is displayed on top. The shifted shortcuts . , and Ret are infrequent, except perhaps if combined with some forms of text-prediction. To illustrate a possible application it is depicted in an editor with @=the cursor and a short margin menu. Bypassing numerical codes as shown in Figure 2 access the margin menu to repeat stored text as by ...--=3=How are you; -----=0 speaks the whole page from 'Hello ' to 'talking '.

.		-

a	e	n	r

i	o	s	t

SP

b c j k

d h

f g p q

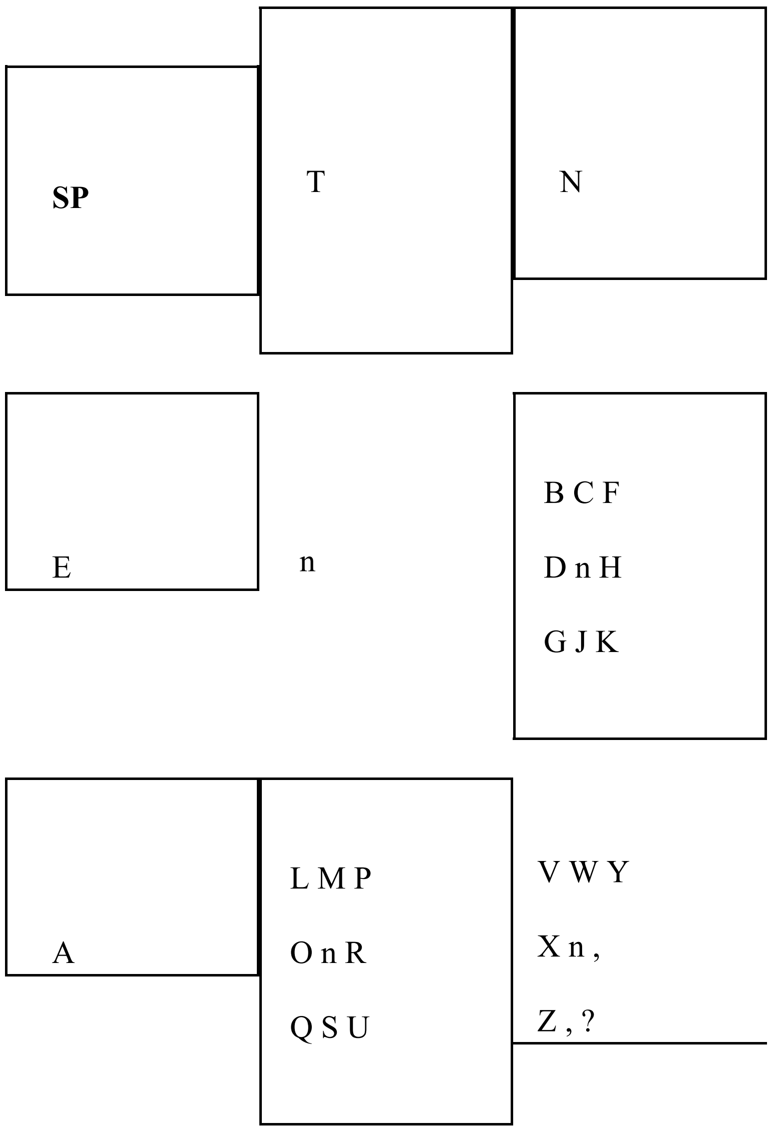
u v y z

l m

w x . ,

;		=

Matrix 16	Matrix 15 may be optimized based on average letter frequencies in English, with some more room for improvement with shifted shortcut. To the author, Matrix 16 still looks rather easy to learn.

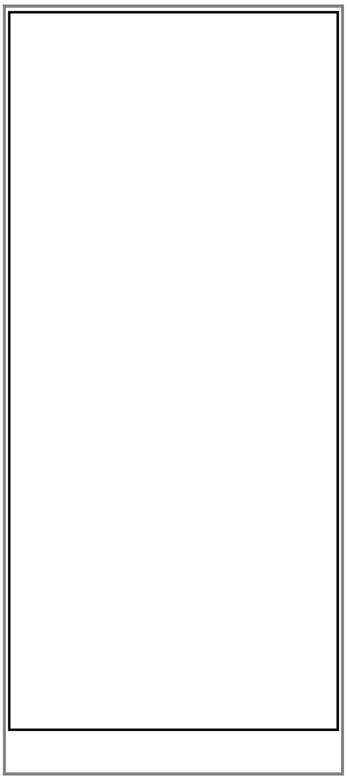


Matrix 2A1 Matrix intended for active scanning with eight-sided joystick, was neither implemented nor tested. With n is meant the joystick in its middle position. To select SP requires only one choice, to northwest, to select B requires two, one to east then one to northwest.

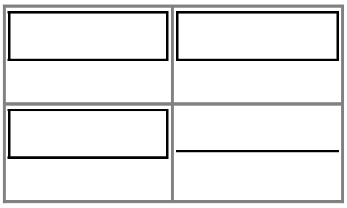
.			-
	a	n	
	o	SP	
..			--

Matrix 2A2 Scan matrix to learn the alphabet or to make rapid selections. With group-wise scanning .=a -=n ..=o and --=SPace. This possibility was implemented but not tested and is vaguely related to so-called pie-menus with the mouse.

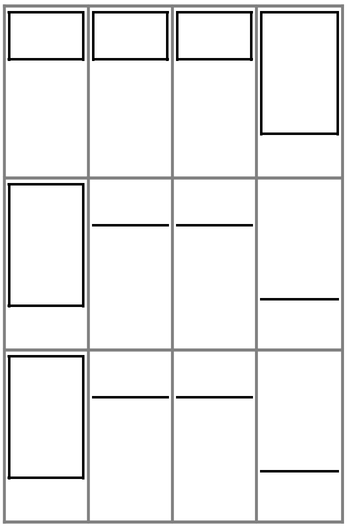
Squares in squares

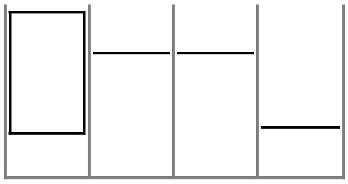


A square



may be divided into four squares ..





that may be divided into four times four squares ..

□	□	□	□	□	□	□	□
□	_	_	_	_	_	_	_
□	_	_	_	_	_	_	_
□	_	_	_	_	_	_	_
□	_	_	_	_	_	_	_
□	_	_	_	_	_	_	_
□	_	_	_	_	_	_	_
□	_	_	_	_	_	_	_

times four is sixty-four squares ...

Figure 1. 4 x 4 x 4 = 64 and therefore three choices out of four

alternatives allow to choose from 64 items.

OVERVIEW OF GROUP-WISE SCANNING WITH TWO SWITCHES

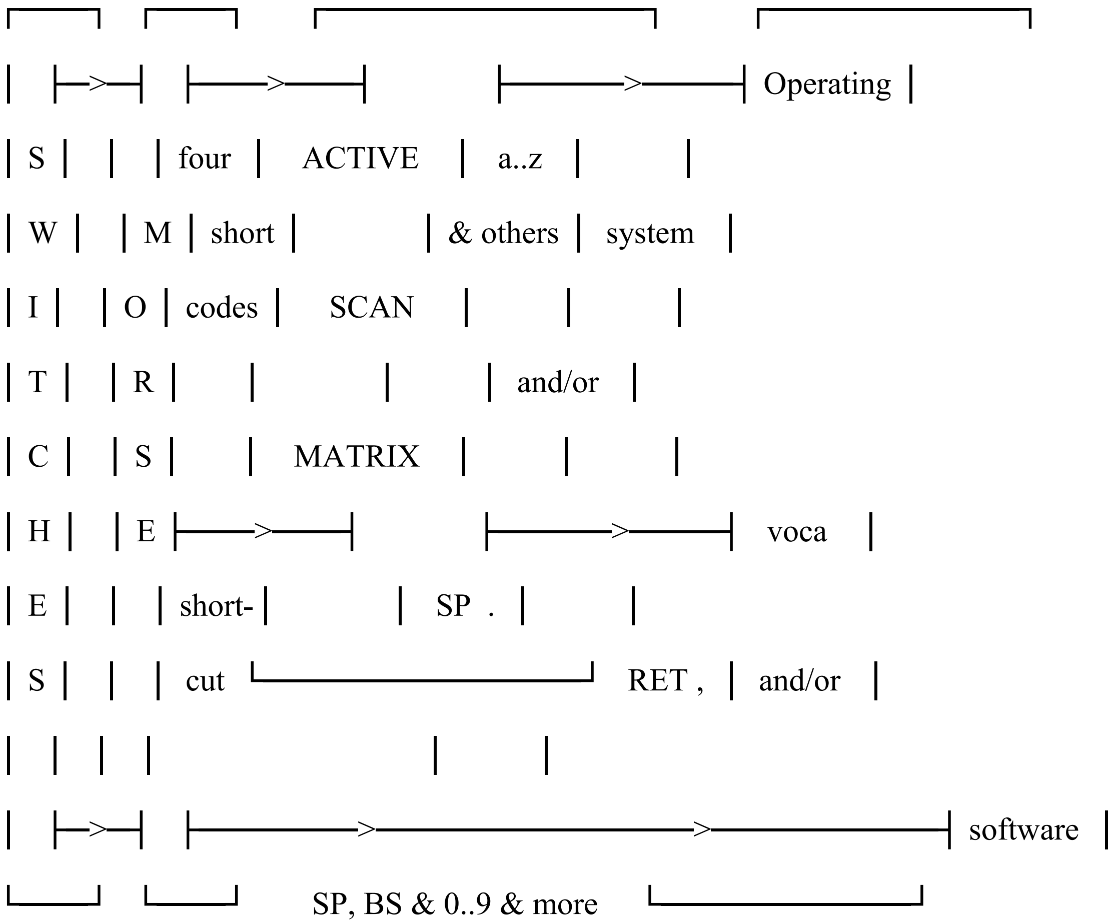


Figure 2 Voca means voice output communication aid. Codes for SP, Backspace (BS) and numerals may bypass the active scan matrix, while the meaning of the (shifted) shortcut is decided by the actual state of the scan matrix.

Rates with simulated passive scanning

	Characters per minute
43-key Linear qwerty	3.89
43-key Linear alphabetical	4.93
43-key Linear letter-frequency	6.73
43-key RC qwerty	8.89
43-key RC alphabetical	10.73
43-key RC letter-frequency	12.53

Table 1 Data computed from Venkatagiri (1999). This simulation-study with scanning pauses of 1 second is the state of the art. As expected for a simulation study no standard deviations (SD) can be computed.

Comparison with pauses of 1000 ms

	Chars/ minute	SD	Clicks/char
Active scanning with a single switch	14.73	±0.48	4.7
Group-wise scanning	15.63	±0.22	3.8

Table 2 Data from an evaluation study with a subset of the same data and with pauses of 1 second as in Venkatagiri (1999). N=1, 1027 chars, four times eight lines. Clicks/char include clicks generated by the machine when a switch was held.

Comparison with pauses of 400 ms

	Chars/ minute	SD	Clicks/char
Active scanning with a single switch pauses = 440 ms	20.44	±0.89	6.06
Optimised group-wise scanning with two switches pauses = 400 ms	25.50	±1.41	2.93

Table 3 Test with optimised group-wise scanning pause at 400 ms and active scanning with a single switch, pauses at 440 ms, Matrices 12 and 3. 901 chars were scanned twice. The high rate of clicks/char with scanning with a single switch was due to cycling of the focus on the first row: 440 ms is a bit too short with active scanning and for this test subject.

	Chars/minute	SD	Clicks/ character
Active cr-scanning with two switches, pauses of 1000 ms A	21.77	±1.15	4.57
Active cr-scanning with two switches, pauses of 400 ms B	28.48	±3.55	4.77

Table 4 659 chars and ABBA design with two different pause times, audible clicks, 4x5 lines and Matrix 11.

Comparison with same phrase

	Chars/minute	SD	Clicks/ character
Passive with a single switch, pauses 1000 ms (C)	12.3	±0.45	3.1
Passive with a single switch, pauses 480 ms (D)	22.1	±2.1	3.2
Active with a single switch, pauses 480 ms (E)	25.1	±1.1	5.1
Active scanning with two switches, pauses 480 ms (F)	30.3	±3.6	5.2

Table 5 A phrase of 25 chars typed over four times with Matrix 10 and conditions C to F, cr-scanning. To correct an error costs clicks; therefore clicks/char are above 3.0, the value computed in the model of Appendix 2.

Effect of empty row on number of errors

	Total number of errors	SD	Clicks	Seconds
Active scanning with a single switch, pauses of 400 ms and Matrix 10 (G)	4	0.8	505	245
Active scanning with a single switch, pauses of 400 ms, same matrix, no empty row (H)	26	8.4	631	325

Table 6 Same phrase 25 chars typed over four times with number of errors counted afterwards: G then H. Though they are practically meaningless standard-deviations (SD) were computed of errors per phrase. In the first row, 100 characters were entered in 245 seconds, 24 characters per minute.

Effect of pauses on optimised group-wise scanning

	Characters per minute	SD	Clicks/ character	Number of errors
Pauses 400 ms (I)	32.81	±1.75	2.89	1
Pauses 1000 ms (J)	22.01	±0.58	2.89	2

Table 7 Same phrase typed over four times with optimised group-wise scanning and Matrix 13, shortcut and pause reduction either with pauses of 400 ms IIII or 1000 ms JJJJ. Clicks/character depend very much on the phrase used.

Overview of different experiments with the author as test subject

Scanning technique pauses/codes per character	Condition	Speed in chars per minute ± SD
Passive cr-scanning with a single switch ±5/3	Same phrase C m10 p 1000 D m10 p 480	 12.3 ±0.45 22.1 ±2.1
Active cr-scanning with a single switch 2/±5	Copy task m3 1000 Table 2 m3 p440 Table 3 Same phrase E m10 p480	 14.73 ±0.48 20.44 ±0.89 ±1.1
Active group-wise scanning, single switch ±2.25/5.6	Same phrase m12 p320 see text	23.4 ±0.45
Active cr-scanning with two switches, audible clicks 1/±5	Copy task A m10 p1000	 21.77 ±1.15

	B m10 p400 Same phrase F m10 p480	28.48 ±3.55 30.3 ±3.6
Group-wise scanning with two switches, shortcut 3/±3	Copy task m6 p1000 Table 2 m11 p400 Table 3	 15.63 ±0.22 25.5 ±1.41
Optimised group-wise scanning with two switches, shortcut, pause reduction ±1.45/±2.25	Same phrase J m13 p1000 I m13 p400 Copy task m14 p280 see text	 22.01 ±0.58 32.81 ±1.75 30.55 ±0.7

Table 8 Comparison between several techniques with the author as test subject. m10=Matrix 10, p480 = pause times of 480 ms, etcetera. Chars per minute vary remarkably: $cpm_I > cpm_F > cpm_E > cpm_D$. To put these data in perspective consider the following. The same subject achieves ±40 characters per minute with two-key Morse plus keyboard macros, ±60 characters per minute with a mouse and on screen keyboard, ±110 characters per minute with single finger typing, ±300 characters per minute maximum speed with ten fingers blind-typing for about 5 minutes and almost 800 characters per minute speaking rapidly.

Comments

It took me about half an hour and some instruction to get used to. Would need training to get fast at it. Looks interesting. (Leonie).
I would rather talk with a computer than listen to it. Annoying sound of soundcard, perhaps you could use it without speech. Windows software looks better. (Arnoud).
I spent the first hour to study the keyboard and get an idea of what a computer is. I needed several hours of service to understand all kinds of details, but I can use it though of course I hope that I will never have to. It sometimes makes me feel impatient and I find difficult to wait when I want to scan rows after scanning columns. (Maartje).
Looks like a small piece of software that you can master rapidly. (Carolien).

Table 9 Comments by several test subjects. Note that their impression of rapid and easy learning is not backed up with data from relevant patient groups.

Part of log file

===== New Session =====

Time: 12 U 33 Min 35 Sec 32 Sec100

Frequencies of characters used:

#8 25 #13 20 #27 2 #32 126   , 6   . 21

` 7   a 41   b 7   c 12   d 19   e 51

f 7   g 17   h 22   i 57   j 2   k 10

l 17   m 17   n 36   o 46   p 5   q 1

r 25   s 27   t 47   u 20   v 8   w 9

x 1   y 21   z 1

	Frequency	Length
Words typed in	126	706

Pauses: 320 milliseconds.

Total Morse length: 2586 . & -

Number of Morse codes: 1653 codes.

Shortcuts: 195 times.

Time: 13 U 1 Min 43 Sec 29 Sec100

=========== End of Session ======

Table 10   Log file with Carolien. #8=BS #13=RET #27=ESC #32=SP

	Morse Code	Active group-wise scanning with M16	Cr-scanning, M2 with empty first row
Switches per character	**2.57**	**3.08**	**3.86**
Pauses per Character	**1**	**2.20**	**1**
Pauses per character with pause reduction	**1**	**1.29**	**1**

Table 2A1. Switches and pauses for the average character computed with weighted frequencies of 'A' to 'Z', spaces were not taken into account because with Morse they are often replaced by a double pause. Frequencies were based on Stower (1817), see Table 2a2.

A 8500	N 8000
B 1600	O 8000
C 3000	P 1700
D 4400	Q 500
E 12000	R 6200
F 2500	S 8000
G 1700	T 9000
H 6400	U 3400
I 8000	V 1200
J 400	W 2000
K 800	X 400
L 4000	Y 2000
M 3000	**Z 200**

Table 2A2. Frequencies of characters 'A' to 'Z' from Stower (1817), the same as used by Samuel Morse; they seem rather close to the relative frequencies in modern English.

Appendix 1. Choice times depend on number of options

According to the law of Hick-Hyman in psychology, choice-time increases according to the formula: t(ime) = a log x, with x the number of items to choose from and a an individually variable constant. This law is related to the law of Fitt that is better known and predicts the time to grasp something as a function of its distance: t(ime) = b log d(istance). If for a certain user we assume 1 second for a yes-or-no choice, the law of Hick-Hyman predicts 2 seconds for a choice out of four items and 3 seconds for a choice out of eight items. If a log 2 = 1 second, a log 4 = a log 2^2 = 2 a log 2 = 2 seconds, likewise a log 8 = a log 2^3 = 3 a log 2 = 3 seconds. Should we choose twice out of four alternatives we estimate 2 a log 4 = 4 a log 2; choosing four times out of two alternatives, likewise a total of 16 items, we estimate 4 a log 2 as well. According to the law of Hick-Hyman three choices out of four are time-equivalent to six choices out of two as $4^3=2^6=64$. Trained behaviour may outperform the law of Hick-Hyman as shown by blind typing or by some amateur pianists. Morse code comes close to a counter example, as trained users seem to plan ahead whole words and execute them automatically. Nevertheless, the law of Hick-Hyman has empirical support and helps to model what scanning users go through.

Appendix 2. Models of scanning speed and of switches per character

Rate is the major figure of merit for scanning input (Damper, 1984). This appendix predicts the time (t) per character of different techniques after sufficient learning and based on duration of pauses (p) and of clicks or, in the case of group-wise scanning, of codes (c). We will assume pauses take 1 second and a click or a code takes 0.333 seconds, to easily compute first estimates. With able bodied test subjects 0.5 second and 0.2 second are more realistic. There are many details that limit the value of such models: other user processes as deciding are not taken into account and will overlap with pauses and clicks at low rates, the role of pauses and codes or clicks is not exactly comparable between different techniques and subtle details in the implementation may influence performance. Still, to analyse experimental data and to understand the differences between techniques, it is hard to overestimate the value of quantitative models.

With classical passive row-column scanning users need one click to start, one to accept the row and one to accept the column, three clicks. The number of pauses to select a character will depend on the matrix used, we will assume four pauses. Once the character is reached a click will select it, we count no more pauses, this waiting time is assumed part of the click. Thus field 1,1 in Matrix 1 is marked as 0 pauses. Then we predict t=4p+3c for the average character, 4p+3c=4x1+3x0.333=5 seconds per character, or 60/5=12 characters per minute (cpm).

With active cr-scanning, a single switch and a matrix with an empty first row, we assume six clicks instead of four pauses. One to start the scanning, one for the empty row and the same four as above for the distance. We compute two seconds for two pauses and two for six clicks, four seconds or t=2p+6c, 60/4=15 cpm. With group-wise scanning a code like . is followed by a pause to determine if the user intends . or possibly .. or even ..... (the Morse code of "5") and group-wise scanning therefore needs three pauses and three codes: t=3p+3c or four seconds per character, 60/4=15 cpm as well. The shortcut .- can be selected by one pause and one code, for this reason group-wise scanning can be somewhat faster than scanning with a single switch with equal duration of pauses. With two switches and cr-scanning we have t=1p+6c, three seconds per character, 60/3=20 cpm.

To model optimised group-wised scanning with pause reduction consider Matrix 14. We will assume frequencies of 15% for space, selected by shortcut, 1c+0p, 45% for the combined frequencies of characters aeionrst, 2c+1p, 20% for bcghlmuv, 3c+3p and 20% for dfjkpqwx, 3c+2p. Return, Backspace, punctuation, y and z are neglected. We note in passing that results must be very different with texts that have other relative frequencies. We compute the time to scan the average character t=0.15*c+0.45*(2c+p)+0.20*(3c+2p)+0.20*(3c+3p) or t(ime) = 2.25 c(odes)+1.45 p(auses). If p=1 and c=0.333 t=2.2 seconds or 27 cpm. A similar reasoning as the one above leads to the formula t=2.25c+2.25p if pauses are not reduced. The difference of 0.8 p will partially overlap with physiological muscle relaxation and rate difference shall probably not exceed 10%. Therefore, pause reduction offers a modest advantage at an uncertain price, as is in accordance with Table 7 where measured rate with J is 22, not 27. With continued training with this phrase and Matrix 15 it was possible to achieve 25 cpm, still below the prediction.

We may now formulate the model:

Passive single switch rc-scanning: t=4p+3c

Active single switch cr-scanning: t=2p+6c

Active two switch cr-scanning: t=1p+6c

Ordinary group-wise scanning: t=3p+3c

Group-wise scanning and pause reduction: t=1.2p+3c

Optimised group-wise scanning: t=2.25p+2.25c

Optimised group-wise scanning and

pause reduction: t=1.45p+2.25c

It follows from this model that, if pause times exceed click-times by so much as 15%, all active scanning techniques are faster than passive scanning. Note however, that individual pauses may be different for different techniques and that fatigue and error rates and many other psychological variables are important too. Training will not by itself help to improve on these limits, determined by the length of pauses, but of course with training pauses may be gradually shortened. If spaces take 15% of a text, pauses are 600 ms and codes take 333 ms, group-wise scanning with shortcut (one code only) bound to space requires 0.85*(3*600+3*333)+0.15*(600+333)=2519 ms per key or 23.8 characters per minute. Still higher rates are computed for active cr-scanning with two switches. If clicks are 300 ms, pauses 600 ms and the average character takes five clicks only (no empty row) we get 2.1 seconds per character or 28.6 characters per minute. These predictions are in line with experimental data of Tables 1, 2 and 4.

With pauses fine-tuned to individual users the rates may rise depending on the technique used as well as on the users. For a given test subject, pauses of 400 ms may be agreeable for active scanning with two switches but difficult and error-prone for passive scanning with a single switch, as was confirmed experimentally, see Table 6. The model cannot predict optimum rates for individual users because user processes are not modelled in sufficient detail. Choices in group-wise scanning, a choice out of four codes of length one or two are not fully comparable to choices in cr-scanning, a choice out of two codes of length one, execution times will vary and individual differences exist in psychological traits as learning, motivation and reaction times (note 8).

The fact that group-wise selection allows users to divert their attention during the scanning process and look elsewhere is not accounted for in the model but may be important for some users. 'The hallmark of an attention-demanding process is that its performance varies with other demands for attentional resources' (Mizuko, Reichie et al., 1994) as can be easily noted by anyone who tries to actually communicate with just a scan matrix, Morse code or a bloc note. This important quality of group-wise scanning might show in a copy task maintained for a long period or combined with other tasks as during a conversation.

The model is not adequate either to predict the effects of combined scanning and word prediction. If with a pause of 1000 ms some subjects profit from word prediction and would be faster than without it, this same word prediction may well cost them time with pauses of 500 ms. With shorter pauses less overlap in time is possible between user processes as wait-for-pause-to-pass, make-choice, look-at-prediction-list, decide-if-prediction-is-desirable and spot-next-character-in-scan matrix. This is in accordance with some of the results reported by Koester and Levine (1994), who compared scanning with scanning plus word prediction integrated in the scan matrix, and found a small (8.7%) and non-significant advantage for scanning plus word prediction. Other explanations are certainly possible to explain those specific results. Still, potential users are well advised to practise scanning before they try to combine it with word prediction. Word prediction need not speed up and may even cost time and, more important to actual users, will cost attention (Note 9).

Besides time to activate switches and pauses needed by the machine, users clearly need time and attention to read, to choose, to coordinate and more. We may add a variable o= other activities to the formulas and write, for optimised group-wise scanning, t=o+2.25c+1.45p. O depends on p and c, as such other activities will overlap with long pauses, and o may be computed retrospectively. According to the last formula we may compute if rate is 30 cpm (t=2000 ms) and p=c=300 ms, then o=890 ms; as

this seems too short for several choices we conclude that training is needed to achieve 30 cpm. In Table 7 and retrospectively o_i=507 < o_j=527 and p_i=400 < p_j=1000. The sum of o_i per minute is more than the sum of o_j per minute as cpm_i=32.8 and cpm_j=22.0. With single switch group-wise scanning, Matrix 12 and pauses of 320 ms the author achieved 23.4 cpm at 5.6 clicks per character. We may now use the formula t=p+5.6c+o and substitute p=320 c=250 (though c was not measured directly) and t=60/23.4 seconds to compute o=844 ms.

Possibly, even faster rates are achieved by users who can freely access a joystick in eight directions, as claimed by the Darci Institute. If combined with Huffman code the formula average time per character = 1.5 c_8 seems possible, with c_8 the time to choose from eight alternatives, see Matrix 2A1. Optimised Darci Too might well be the fastest system with unlimited training, and for a very limited patient group. Regrettably, there seems to be no study yet to investigate it. Something much like a joystick is possible with active group-wise scanning, see Matrix 2A2.

(* Matrix 2A1 and 2A2 about here *)

Another way to compare different scanning techniques is to forget about the users and compute the number of switches and pauses needed for the average character, weighted for frequency. If character c has frequency f_c and the number of switches for this character is s_c we may compute switches per char as (Sum $f_c.s_c$)/(Sum f_c). Results are as in Table 2A1, computed with the same frequencies of 'A' to 'Z' as Samuel Morse used in 1817, and that are rather close to current day English. See Table 2A2. If the numbers for switches and pauses per character are added up the results vary from 3.57 for Morse, 4.40 for group-wise scanning with pause reduction, 4.86 for cr-scanning with empty row to 5.31 for group-wise scanning without pause reduction. Average switch counts per character can also be computed based on single phrases and are then highly variable, especially if spaces are taken into account. This is good reason to be careful with interpretation of results from small sample sizes, including results in Tables 3 to 7!

(* Tables 2A1 and 2A2 about here *)

Appendix 3 Shareware version

Caregivers are advised to learn active scanning themselves before they offer it to clients. The software used above may be requested by all readers for use in research, trade, training, patient care or development. It runs under Dos (or in a dos-box under Windows), can be combined with external synthetic speech and is integrated in a simple editor with margin menu and paragraphs to store and to retrieve small stories. Text prediction and abbreviation expansion may be activated as well as the ability to create log files. Switches can be read from an opened keyboard and it offers Morse access with a special function called repeat-it plus nine different scanning techniques. The techniques described above are present but also new techniques as single key group-wise scanning, stepped group-wise scanning, Morse based cr-scanning and cr-scanning with a third switch to accept.

Repeat-it in Morse resembles the sticky arrow feature described with scanning. Arrows and cursor movement keys can be repeated with a series of dots and stopped with dash and with all other codes. The code temporarily and automatically changes meaning, therefore ¬¬¬¬¬ or five times Arrow Up (.-..- or a&u combined) then five times Arrow Left (.-.-.. or a&l combined) can be selected with .-..- .... .-.-.. .... nineteen switches for ten arrow keys. The series of dots repeat.

An alternative is to use a set of macros in visual basic or to ask local software providers to add two-bit quartering to their products.

Acknowledgements

The author gratefully acknowledges helpful criticisms from John Todman of Dundee, Scotland, Europe and from Horabail Venkatagiri of Ames, Iowa, United States of America.

About the author

Joris Verrips works as a medical doctor in the field of occupational medicine. In the past, he has also worked as a teacher of mathematics, as a computer programmer and as a text processor. He is interested in AAC-research with text based user interfaces since many years.