DASHER – An Efficient Writing System for Brain Computer Interfaces?

Sebastian A. Wills and David J. C. MacKay

IEEE Trans. Neural systems and Rehabilitation Engineering. (2006.06)

Presenter : Soogil Woo
GIST, Dept. of Information and Communication, INFONET Lab.

Gwangju Institute of Science and Technology

Background

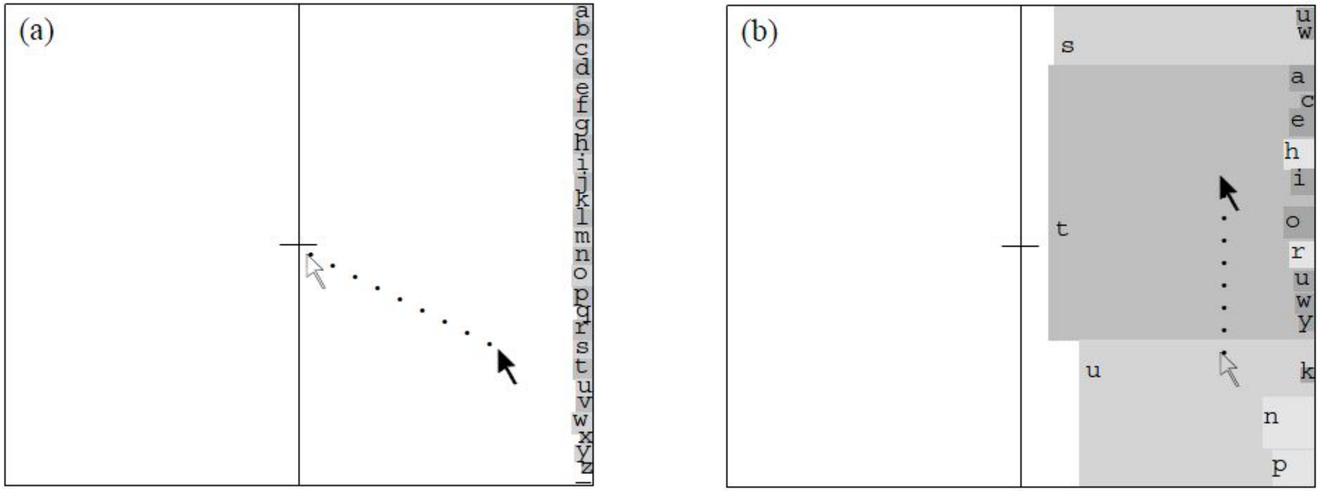
- The P300 (P3) wave is an event related potential (ERP) component elicited in the process of decision making.
- In neurology, Steady State Visually Evoked Potentials (SSVEP) are signals that are natural responses to visual stimulation at specific frequencies. When the retina is excited by a visual stimulus ranging from 3.5 Hz to 75 Hz, the brain generates electrical activity at the same (or multiples of) frequency of the visual stimulus.
- Dasher is a computer accessibility tool which enables users to write without using a keyboard, by entering text on a screen using a pointing device such as a mouse, a touchpad, a touch screen, a roller ball, a joystick, a Push-button, or even mice operated by the foot or head.

Introduction

- DASHER is a human computer interface for entering text using continuous or discrete gestures.
- DASHER efficiently converts bits received from the user into text, and has been shown to be a competitive alternative to existing text-entry methods in situations where an ordinary keyboard cannot be used. (PDAs, mobile phones, for handicapped person)
- DASHER is free, open-source software.
- In DASHER, the size of each box within its parent box is determined by the corresponding letter's probability according to a language model.
- As result, sequences of characters that are well predicted by the language model take less time to zoom into.
- Improbable sequences of characters are always possible to write, but take longer.

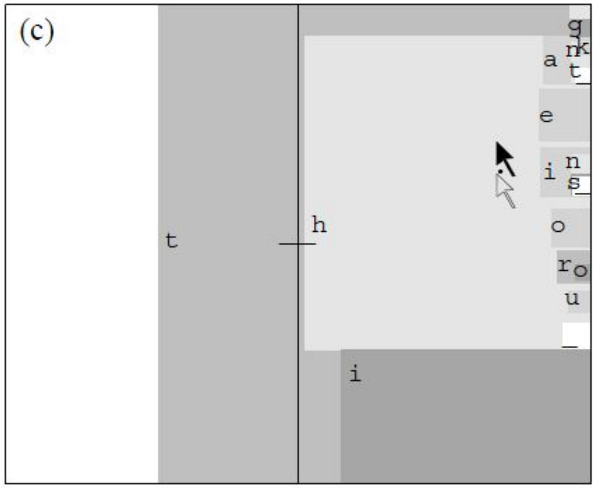
Introduction & Overall Users' View

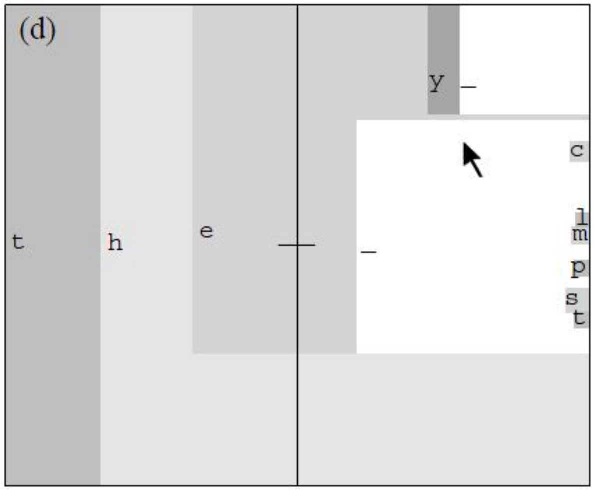
- They first describe how Dasher is used to enter the word 'the'.
- Figure (a) show the initial configuration, with an alphabet of 27 characters displayed in a column.
- There are 26 lower case letters and the symbol '_'(under bar) represents a space.
- The user writes the first letter by making a gesture towards the letter's rectangle.
- The trails show the user moving the mouse towards the letter 't'.



Introduction & Overall Users' View

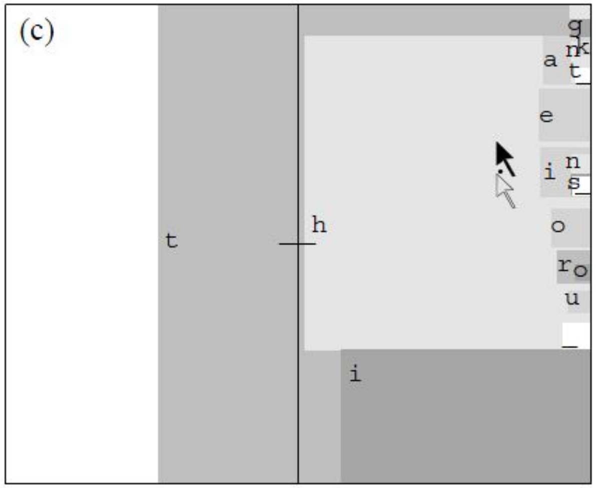
- The point of view zooms towards this letter (figure (b)).
- As the rectangles get larger possible extensions of the written string appear within the rectangle that they are moving towards.
- If they are moving into the 't', rectangles corresponding to 'ta', 'tb',…, 'th',…,'tz' appear in a vertical line like the first line.
- The heights of the rectangles correspond to the probabilities of these strings, given the languages.

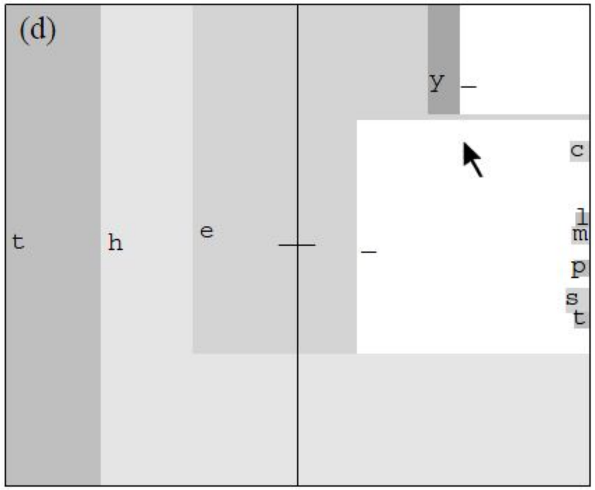




Introduction & Overall Users' View

- In English, 'ta' is quite probable; 'tb', is less so; 'th' is very probable.
- It is easy to gesture our point of view into 'th' (figure (c)), and from there into 'the' (figure (d)).



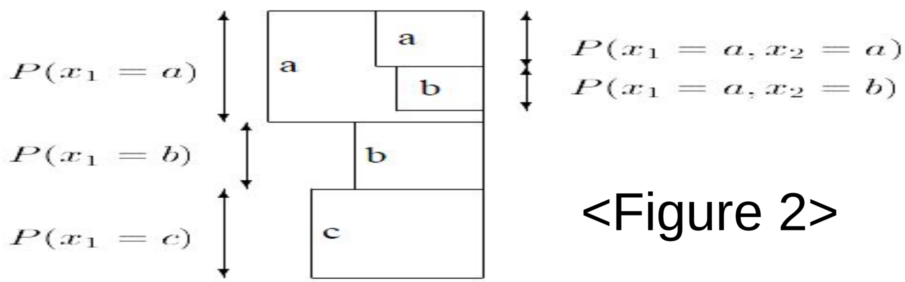


Introduction & Probabilistic Model Determines

- In a given context, they display the alphabet of possible continuations as a column of characters as shown in figure 2.
- The division of the right-hand vertical is analogous to arithmetic coding.
- Let their alphabet be $A_X = \{a_1, a_2, ..., a_I\}$.
- They divide the real line[0,1) into I intervals of lengths equal to the probabilities $P(x_i = a_i)$.
- They subdivide the interval a_i into intervals denoted $a_i a_1, a_i a_2, a_i a_3, ..., a_i a_I$ such that the length of the interval $a_i a_j$ is

$$P(x_1 = a_1, x_2 = a_2) = P(x_1 = a_i) P(x_2 = a_j \mid x_1 = a_i)$$

- They use a language model. (PPM and PPM5D+)



<Figure 2>

Input Devices currently used with DASHER

A. 2-D Continuous Input

- DASHER was first developed to be driven by continuous two-dimensional (2-D) gestures by directly controlling the position of a pointer on the screen.
- If the user moves the pointer away from the origin, the interface zooms in towards the location pointed to by the vector from the origin to the pointer.
- 2-D input devices used with DASHER include mouse, touch-screen, gaze tracker, and head mouse.
- Under mouse control, novice users can reach writing speeds of 25 words/min after 1 h of practice; expert users can write at 35 words/min.
- Under eye control alone users familiar with DASHER can write at 25 words/min, faster than any other gaze-writing system they are aware of.

Input Devices currently used with DASHER

A. 2-D Continuous Input (Example)

Input Devices currently used with DASHER

B. 1-D Continuous Input

- For input devices offering a single continuous dimension of control, DASHER maps the one-dimensional (1-D) input onto a continuous curve within the normal 2-D control space.
- Midrange values of the input control the direction in which to zoom.
- Values towards the extremes of the available range allow the user to zoom out and pause the interface.

C. Discrete Inputs

- Users who can activate buttons (virtual or physical) but cannot reliably provide a continuous output can use one of DASHER's "button modes". (The direct 2-button mode maps 1 button to the action of zooming in one the top half of the visible DASHER land-scape, and the second button to zooming in on the bottom half.)

Input Devices currently used with DASHER

C. Discrete Inputs

- In button mode, DASHER converts bits from the user into written text at exactly the compression rate achieved by the language model.
- DASHER's current language model PPMD5 compresses English text to around 2 bits per character.
- DASHER outputs one character for every two bits provided by the user's button presses.

DASHER as a BCI user interface

A. Motivation

- Current brain computer interface (BCI) systems extract data from the user at a lower information transfer rate than typical physical user interfaces.
- DASHER offers an efficient method for converting the output of a BCI into text.
- DASHER can also use information about the reliability if the signals generated by the user.
- DASHER's language model can be initialized using text that is biased towards a limited set of phrases and words that the user is likely to wish to communicate.
- The user will be able to write these phrases, or variants of them, extremely quickly, while retaining the ability to write any other phrase should they wish to.

DASHER as a BCI user interface

B. Continuous Control

- Many BCI systems output a continuous 1- or 2-D signal which could be used to drive DASHER directly.
- DASHER is well-suited to a BCI signal which is likely to be under imperfect control of the user.
- In DASHER, users write by navigating to what they want to say, not by selecting letters or words.
- If the user accidentally steers in the wrong direction, they can correct their mistake by subsequent compensatory action.
- As with all navigation, all that matters is the final location arrived at.

DASHER as a BCI user interface

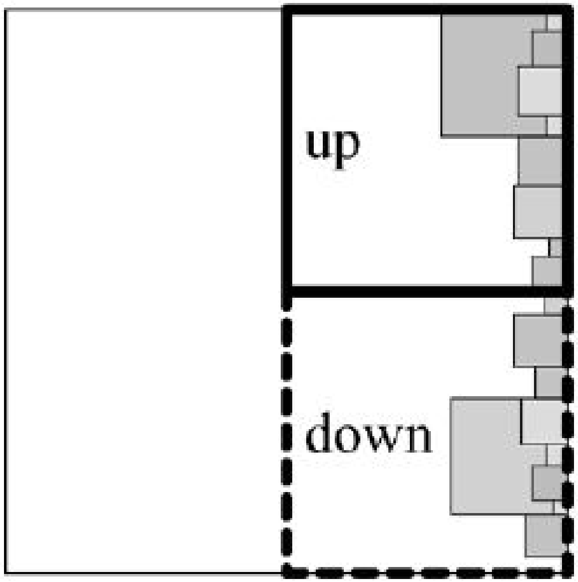
C. Discrete Control

- BCI systems that emit discrete events fall into two categories.
- The first category contains systems which internally convert a continuous variable into discrete outputs
- Systems in which the user makes selections by driving a cursor to one of two or more on-screen targets fall into this category.
- They suggest that the best strategy for using DASHER with BCI systems in this category may be to use the continuous variable to drive DASHER directly, without an intermediate conversion into discrete options.
- The second category contains systems which are intrinsically discrete in nature.
- For example, both P300 and steady-state visually evoked potential (SSVEP) interfaces determine which of several discrete visual targets the user is attending to.

DASHER as a BCI user interface

C. Discrete Control

- A natural way to use these techniques would be to paint P300 targets or SSVEP regions onto the DASHER landscape.
- For example, in the case of SSVEP, the right-hand half of the DASHER landscape could be covered by two or more regions flickering at different frequencies (Fig. 3).



Depending on which region the user attends to, DASHER zooms in on the top or bottom half of the screen. <Figure 3>

DASHER as a BCI user interface

C. Discrete Control

- To zoom into one of these regions, the user attends to that region.
- The BCI system detects which region the user is attending to and causes DASHER to zoom in appropriately, and the cycle repeats. Likewise, P300 targets could be arranged down the right-hand side of the DASHER landscape, instead of in the commonly used speller grid.
- If the accuracy of the BCI system is high, then the optimal strategy for dealing with the rare errors that do occur may be to simply provide an additional target which instructs DASHER to undo the previous action (i.e., zoom out).
- Such a target needs to be present anyway, in case the user makes a mistake in selecting which region of the DASHER landscape contains the text they are trying to write.
- This is similar to the strategy of adding a “delete” node to a binary decision tree.

DASHER as a BCI user interface

C. Discrete Control

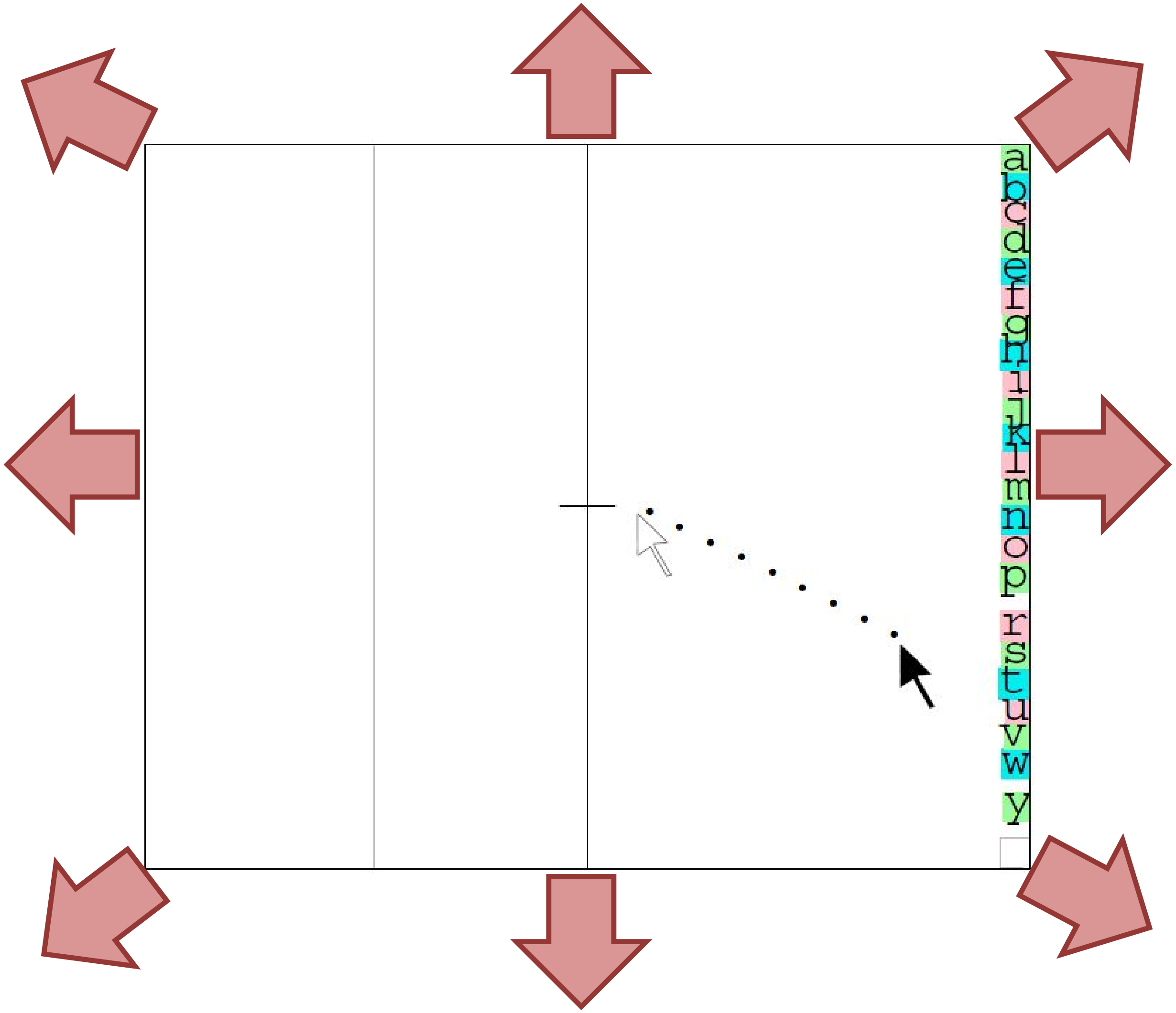
- However, if the BCI misclassification rate is high, they suggest that the optimal strategy is to model the BCI system as a noisy communication channel between the user and the computer, and to use information theory to inform the choice of an error-correcting code to use.
- For example, instead of accumulating evidence that the user is attending to a particular target over a single, long trial, it may be more efficient to run several shorter trials, each one individually less reliable.
- By varying the SSVEP frequencies on each target in each trial according to the coding scheme specied by the error-correcting code, the overall information transfer rate may be improved.

Conclusion

- They wish to make the best possible use of the bits of information content that can be generated by severely disabled people.
- DASHER offers a paradigm for efficiently converting these bits to communication symbols.
- DASHER has proved its effectiveness for people able to use a gaze tracker or make other motor actions.
- DASHER will be equally useful to users who retain functioning vision but are limited to communication through a BCI.

Idea

- DASHER + SSVEP (Continuous Control) (Mouse Control)



Thank you!