Efficient Communication Through The Timings of One Or Two Buttons

JAMES M. G. MEAD
and
PHILIP J. COWANS
and
DAVID J. C. MACKAY
Department of Physics, University of Cambridge
Cavendish Laboratory, J. J. Thomson Ave., Cambridge, CB3 0HE, UK

Dasher is a communication system in which writing is a navigational process. Here we evaluate versions in which navigation is controlled through the timing of presses of one or two buttons. Using the two-button mode, after 1 hour of practice, novice users can write at up to 14 words per minute with virtually no spelling mistakes. Experts, with over 5 hours of experience, can write as fast as 25 words per minute. Preliminary results are presented for the one-button mode, novice users achieving a maximum writing speed of 17 words per minute. We conclude that both these versions of Dasher are gesture-efficient communication systems, ideal for users for whom every gesture is an effort.

Categories and Subject Descriptors: []:

General Terms:

Additional Key Words and Phrases: Information theory, assistive technology, Dasher

1. AN INTRODUCTION TO DASHER

Dasher, introduced in [Ward and MacKay 2002] and [Ward et al. 2002], is a communication system allowing efficient text entry in situations where a traditional keyboard cannot be used, for example as an accessibility tool or for mobile devices where size is a limiting factor. Dasher expresses writing with a navigational metaphor. Any piece of text can be selected by going into a library that contains all possible books and finding the book that contains exactly that text. What is written depends on where the user goes. In Dasher's idealised library, the 'books', represented as a sequence of nested boxes, are arranged alphabetically on one shelf. The writing process is made efficient by the use of a language model, which predicts the probability of each letter in a given context and allocates shelf space accordingly. Specifically, the language model used by Dasher is a variant of Prediction by Partial Match (PPM), whose performance is comparable to the state of the art in the field of text compression [Bell et al. 1990]. A picture of the Dasher interface while writing the word 'hello' is shown in Figure 1(a). Dasher is distributed freely as open source software. We encourage readers to visit `http://www.dasher.org.uk/` and try Dasher for themselves.

Dasher is based on ideas from information theory, particularly inverse arithmetic coding [Witten et al. 1987] [MacKay 2003]. As a result we believe that Dasher is near-optimal in its conversion of information from one form to another. The

...

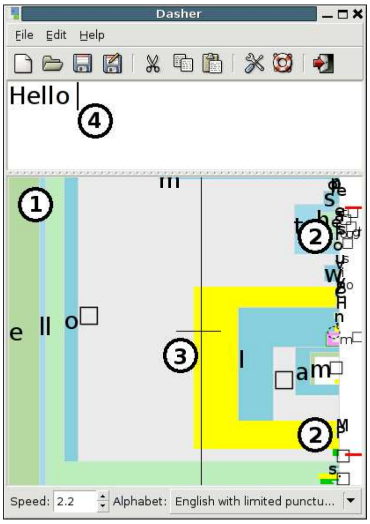


Fig. 1: The Dasher display illustrating the two-button mode. The main display area (1) shows text as a sequence of nested boxes, which when active zooms towards the midpoint on the right hand side. Button presses cause the display to jump so that the point aligned with one of the two fiducial markers (2) is moved to the centre. When a box crosses the centre marker (3), the corresponding symbol is added to the edit area (4).

approach taken decouples the issues of efficient information-capture and efficient language-generation. Unlike in most interfaces, a Dasher user's gestures have no direct relationship to particular symbols in the language.

Dasher was originally designed to be operated through continuous gestures made using a pointing device such as a mouse, track-ball or gaze tracker. Writing speeds of up to 34 words per minute using a mouse [Ward et al. 2002] and up to 25 words per minute using a gaze tracker [Ward and MacKay 2002] have been reported. Dasher has also been driven by head mouse and breathing [Shorrock et al. 2005].

MacKay et al. [2004] described methods for steering Dasher by buttons, mainly focusing on a user model that assumes that the user can choose between a small number of buttons (between 2 and 4), but that the timing of the button presses cannot be used to convey information. In this paper we describe two new modes of Dasher that allow navigation through the *t*iming of button presses. We describe detailed results for a two-button mode, and present preliminary results for a one-button mode. Both modes are aimed at people for whom every press is an effort – they are able to make precise presses, but perhaps only at a low frequency. Precise presses are rewarded, as they convey a lot of information. An underlying theme of this paper is the ability of button Dasher to capture a large amount of information per gesture.

The remainder of this paper is structured as follows: In Section 2 we describe the new modes, and analyse some of their properties. Sections 3 and 4 contain results of user trials for the two-button mode, and Section 5 contains results of trials using the one-button mode. Section 6 summarises our findings.

2. DYNAMIC BUTTON DASHER

This section introduces the new button modes which will be evaluated later in this paper. Before giving details, we first motivate the design of these modes from a theoretical viewpoint. We also introduce a model of the frequency of presses in one of the modes which will be used to analyse experimental results.

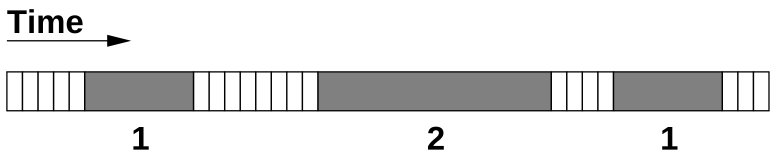


Fig. 2: A schematic representation of the relation between the timing of button presses and symbol transmission. Time is divided into intervals, during which the user chooses either to press one of the buttons (shaded boxes labelled 1 and 2) or to do nothing (empty boxes). Each action has a duration: doing nothing has the shortest duration; pressing button 1 takes longer, and pressing button 2 takes longest. The resulting sequence may be regarded as a message communicated by the user to the computer.

2.1 Motivation

Consider the problem of communicating by transmitting symbols over a reliable channel. The sender has a free choice among N alternative symbols, and the n^{th} symbol takes time takes time τ_n to be transmitted. We would like to find the probability distribution over symbols, $\boldsymbol{p} = (p_1, \ldots, p_N)$, that maximises the rate of information transmission, R,

$$R = \frac{\sum_n p_n \log_2 \left(\frac{1}{p_n}\right)}{\sum_n p_n \tau_n} \tag{1}$$

The optimal value of $\boldsymbol{p}$ satisfies the self-consistent relation

$$p_n = 2^{-R\tau_n} \tag{2}$$

from which the information content of symbol n, h_n, can be derived:

$$h_n = \log_2 \frac{1}{p_n} = R\tau_n \tag{3}$$

As this is proportional to τ_n, it follows that in order to communicate optimally, the mean rate at which information is produced during the transmission of each symbol should be the same for all symbols.

The ability to communicate information by timings of button presses may be related to this scenario by interpreting each button press as the transmission of a symbol. We assume that there is a recovery time resulting from a mixture of mechanical and cognitive factors associated with each button, and this corresponds to the appropriate transmission time. The action of 'not pressing a button' may also be regarded as a symbol, with a briefer duration representing the timing accuracy. This process is represented schematically in Figure 2.

In Dasher we associate all actions (including the decision not to press a button) with navigational events, zooming the display by some factor, the base-two logarithm of which is the amount of information generated by an action. Equation 3 suggests that we should associate navigation events with button-presses so that all actions generate information at the same rate. This insight motivates the idea that Dasher should zoom continuously at a steady rate, and that button presses should simply change the point about which the zooming occurs.

2.2 Two-Button Mode

In two-button mode, Dasher zooms continually towards the centre of the display. When the user presses a button the display is offset either upwards or downwards by a fixed amount depending on which button is pressed. There are two fiducial markers (horizontal red lines – they can be seen in Figure 1), which indicate the points to which the centre will jump. The two buttons control which fiducial is selected; one is associated with each fiducial. To write a message (which corresponds to a tiny point on the alphabetically-ordered Dasher shelf), one first waits for the view to zoom until that desired point is adjacent to either of the fiducials, at which moment the relevant button should be pressed. After the subsequent vertical shift has occurred the display will continue to zoom and this process is repeated until the desired text has been entered.

A Press Rate Model for Two-Button Mode. This section describes a theoretical model for the rate of button presses as a function of information entry rate and timing accuracy. In the following the co-ordinate origin is taken to be the centre of the screen, with positive coordinates being towards the top. We define τ_e to be the e-folding time, i.e. the time required to zoom in by a factor of the natural number, e.

To write a message, the user aims to press one of the buttons when the fiducial is adjacent to the desired point on the screen. If the user were capable of perfect accuracy then a single button press would be all that is required, as Dasher would subsequently zoom towards the centre of the display until the desired text was entered. However, in practice there will be an error in the timing of the press, which will result in a spatial offset as the display zooms. Let ϵ_0 be a random variable denoting this distance immediately after the button is pressed. As Dasher continues to zoom the distance will grow exponentially so that after a time, t, the vertical distance between the centre of the display and the destination will be

$$\epsilon(t) = \epsilon_0 e^{t/\tau_e} \tag{4}$$

Suppose at time T the fiducial, located at coordinate ϕ, is aligned with the desired location, so $\epsilon(T) = \phi$ and hence,

$$\phi = \epsilon_0 e^{T/\tau_e} \tag{5}$$

As the display zooms, the speed at which the target location moves is given by

$$\frac{\partial \epsilon}{\partial t} = \frac{\epsilon(t)}{\tau_e} \tag{6}$$

In other words, a timing error of δt in the button press will result in a spatial error of

$$\epsilon_0 = \delta t \frac{\partial \epsilon}{\partial t} = \delta t \frac{\phi}{\tau_e} \tag{7}$$

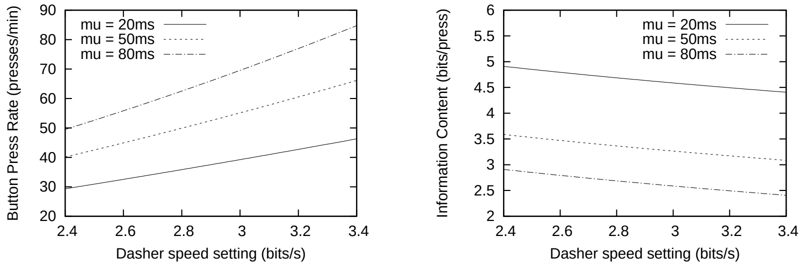
By assuming a uniform distribution for δt, we can find the mean rate of presses, denoted $\langle \Gamma \rangle = \frac{1}{\langle T \rangle}$, at a given zooming speed,

$$\langle \Gamma \rangle = \frac{1}{\langle T \rangle} = \frac{1}{\tau_e \left(\ln\left(\frac{\tau_e}{\mu}\right) + 1 \right)} \tag{8}$$

Where $\mu = \langle |\delta t| \rangle$ is the average absolute timing error. This expression is plotted for a variety of μ in Figure 3(a). Figure 3(b) shows the corresponding average information content per button-press.

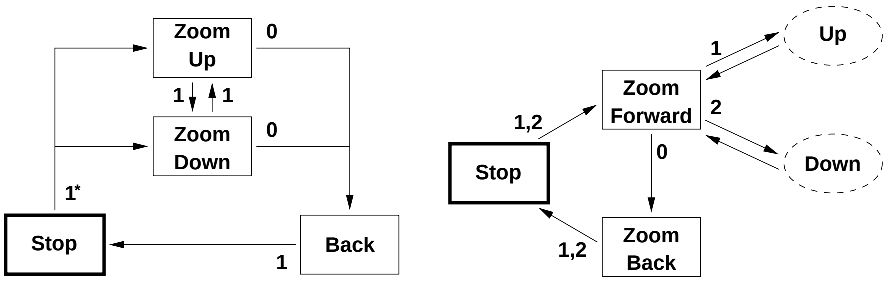
2.3 One-Button Mode

When only one button is available, a different approach is required. In one-button mode Dasher zooms continuously towards one of two fixed points located near the top and bottom of the display, with the action of the button being to switch between them. To enter a message, the user presses the button when the desired text approaches either of these points, ensuring that it remains within the accessible area of the display.



(a) (b)

Fig. 3: Modelled rate of presses, expressed as (a) presses per minute and (b) bits per press, derived using (8) for different values of μ.



(a) (b)

Fig. 4: State diagrams for (a) one-button mode and (b) two-button mode. Initial states are shown in bold. Dashed ellipses represent transient states, i.e. those which are left as soon as they are entered (without another button being pressed). Numbers represent the button which is pressed to activate the corresponding transition, with 0 representing a special action such as multiple presses of any button, a long press of any button, a timeout event or an additional physical button if present. Note that the transition out of 'Stop' in one-button mode will return to the most recently active state out of 'Up' and 'Down', with this storage not represented on the state diagram for simplicity.

2.4 Error-Correction, Starting and Stopping

As well as controlling navigation, the user must start and stop Dasher and occasionally correct errors. Error correction is achieved by putting Dasher into reverse and unzooming until the correct target is back on the screen. In our implementation zooming is started with a single button press. Unzooming is initiated by one of two methods: either a single 'long hold' (the length of which is configurable) or multiple presses (for example, two) within a specific time period. Again, the number of presses and the time in which they must occur are configurable by the user. Unzooming is also automatically initiated if the user doesn't press any button for a long time. In the experiments described below the multiple press method was always used. Finally, Dasher is stopped by any button press whilst unzooming. This process can be represented by the state diagram shown in Figure 4.

2.5 Automatic Speed Control

In both modes the speed at which zooming occurs is a constant which must be specified. Our experience shows that in many cases a speed which is appropriate most of the time can be overwhelming if the user temporarily loses concentration, or makes a small error in button timing. In some cases the result is a cascade of inaccurate presses as the user repeatedly attempts to compensate for previous errors.

To help users in this situation we developed an automatic speed control which is able to temporarily reduce the speed when the user is judged to be in distress. Specifically, this is triggered when rapid presses occur – the speed control slows the speed whenever Dasher records an interval between subsequent button presses, Δt, below a specified fraction of the median. By slowing the speed in this way, the user has time to make the next press accurately and recover.

3. TWO-BUTTON USER TRIALS ON NOVICE SUBJECTS

The first set of user trials aimed to test how novice users improved in using two-button mode during their first hour of practice. Two male and one female subject were used. None had experience with two-button Dasher. Subject 2 had limited experience with Mouse-Dasher. All had vision corrected to normal and were right-handed. The task was to enter text dictated from Jane Austen's *Emma*[1], a task similar to that used in previous trials [Ward and MacKay 2002]. Dasher's alphabet consisted of lower case and capital letters, the space character and a full stop. Subjects were asked to capitalise words correctly, e.g. 'Mr Knightley'. The language model was trained on text from the same source, but not including any of the passages entered by the users.

The experiments were conducted using standard PC hardware, with a 17" CRT monitor positioned on a desk in front of the user. The Dasher display was of size 700×491 pixels (approximately 175×115mm) and the font size was set to a height of 12 pixels. Input was via two large, circular buttons, approximately 5cm in diameter, which were operated with one hand. Dasher was otherwise configured using the default settings.

The protocol is similar to that in [Ward and MacKay 2002]. The subjects used Dasher in 6 sessions, with two 5 minute periods of writing within each session. Within these 5 minute periods the subjects were read sentences, and entered the text they heard using the Dasher interface. The dictation passages were stored as audio files on a computer and delivered by an experimenter one phrase at a time. Before the first session, subjects read an information sheet explaining the Dasher concept, and were allowed to experiment with mouse-driven Dasher for 5 minutes, and two-button Dasher for a further 5 minutes to familiarise themselves with the controls. The initial speed of Dasher was set to 1.0 bits per second (bps), and before each dictation, the subjects were also allowed to read a copy of the dictation passage, to minimise errors arising from Austen's unusual writing style and spelling. At the end of each of these 5 minute periods, the subject was given the option of increasing/decreasing speed by 0.1, 0.2 or 0.3 bps. After each 5 minute writing period the subjects had a break of 5 minutes. No more than two sessions took place on a single day and the maximum spacing between any two sessions was two days.

3.1 Results

Figure 5 shows the results for the novice subjects. The number of words written is defined to be the number of characters divided by five, and the error rate is the fraction of words misspelled. The most common form of error was the omission of full stops, which users found difficult to locate, although with practice all subjects learnt to write very accurately, with all error fractions below 5% after 1 hour.

4. TWO-BUTTON EXPERT USERS WITH AND WITHOUT AUTOMATIC SPEED CONTROL

In addition to the experiments described in the previous section, we also performed trials on two expert users, both with over 5 hours of experience of two-button mode. One expert underwent four 45 minute sessions; two with the automatic

[1]Available from Project Gutenberg at `http://www.gutenberg.org/etext/158`

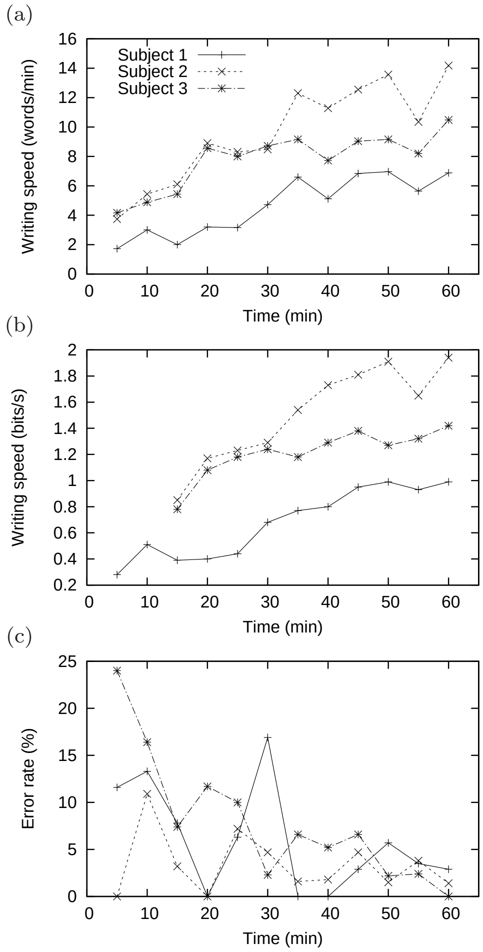


Fig. 5: Text entry rates for novice users as a function of total time spent using the system. Figures show (a) entry rate in words per minute, (b) information rate and (c) fraction of words misspelled.

speed control (ASC) and two without. The other expert performed two sessions, one with the ASC and one without. Sessions consisted of 9 five minute dictations using the same apparatus as before. Speed was incremented by 0.1 bps from 2.6 bps to 3.4 bps between each session. The ASC slowed the speed by 90% whenever a Δt below half of the median value, or a Δt less than or equal to 0.3 s was detected. The latter condition was imposed to allow for physical limitations in the rate at which the buttons could be pressed. After slowing, Dasher would then gradually accelerate back to its original speed over 1 second. Dasher measured the Δt's as it ran, obtaining the median from the first 100 Δt's.

4.1 Results

Writing data is plotted in Figure 6. For expert 1, the mean and standard error for the two sessions is reported. The error rate is not plotted, as this was negligible for both experts.

The fraction of reverses which were preceded by a Δt of less than half the median (without the ASC) is plotted in Figure 7. The average number of reverses used at each speed is plotted in Figure 8.

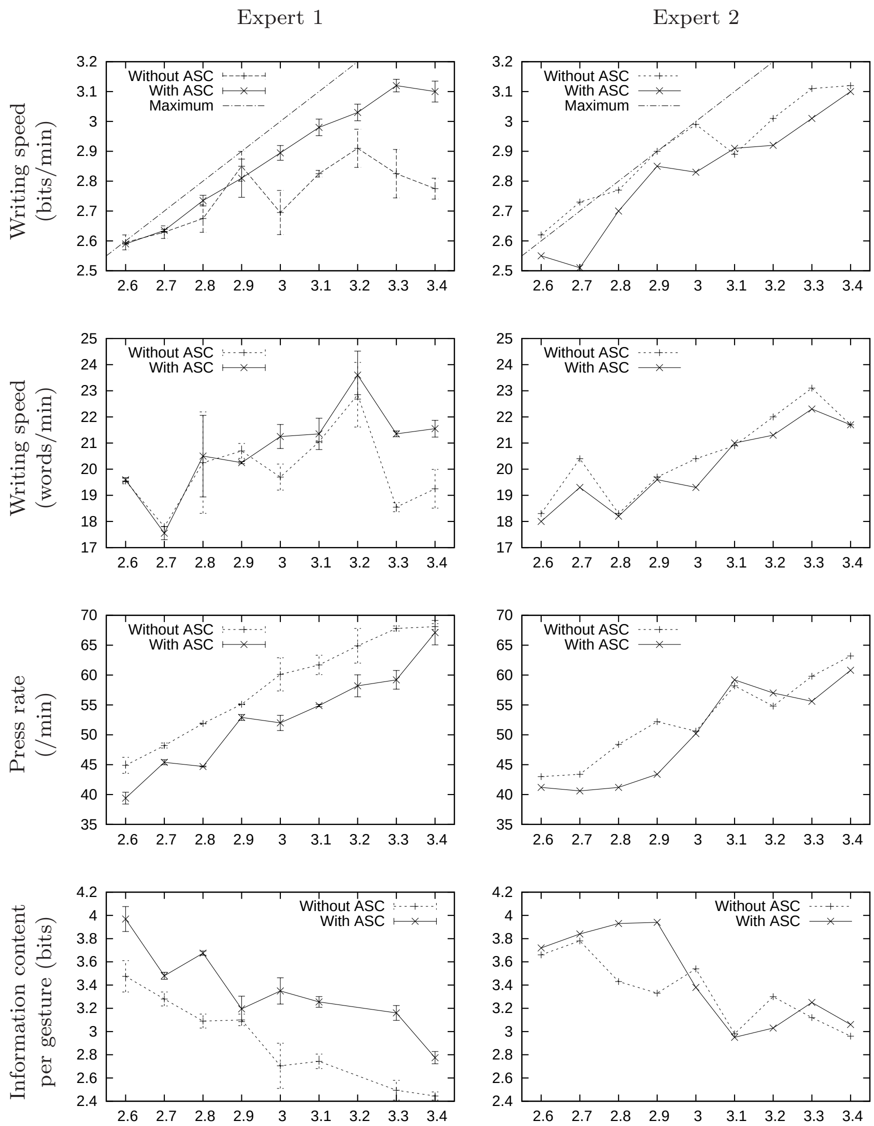


Fig. 6: Results for two-button mode as a function of the imposed zooming speed (horizontal axis) for expert users with and without the automatic speed control (ASC). The top two charts also show the 'maximum' rate of writing, i.e. that which would be achieved if Dasher was never stopped or put into reverse. This is occasionally exceeded as a result in fluctuations arising from the discrete nature of text.

4.2 Analysis

During these trials expert 1 reported the most comfortable speed to be at 3.0 bps, while for expert 2 this was at 2.8 bps. In both cases these speeds are below the point at which maximum writing speeds were obtained according to Figure 6, indicating that there is a trade-off to be made between speed and user comfort.

The ASC has almost eradicated the use of the reverse by expert 1 (Figure 8). This is an important feature of the ASC if two-button mode is to be used by users with disabilities, at whom it is aimed – for many, the long hold or double-click could be a difficult gesture to make, and so cutting down the number of such gestures is desirable.

Figure 6 shows that the ASC improves the information rate and bits conveyed per

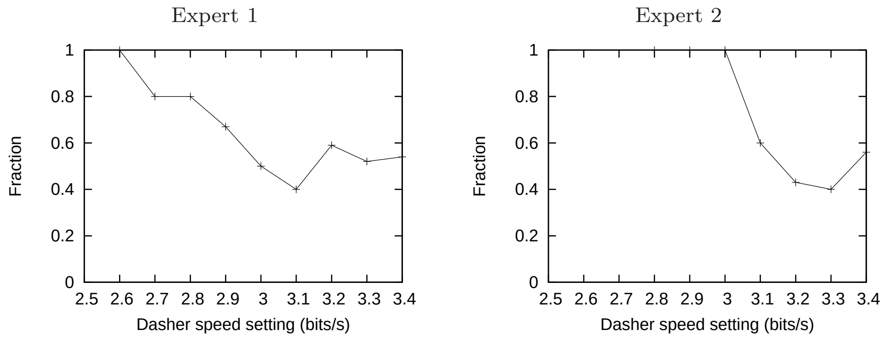


Fig. 7: Fraction of reverses without automatic speed control preceded by an interval between presses of less than half of the median. In other words, the fraction of presses for which the automatic speed control would have been activated.

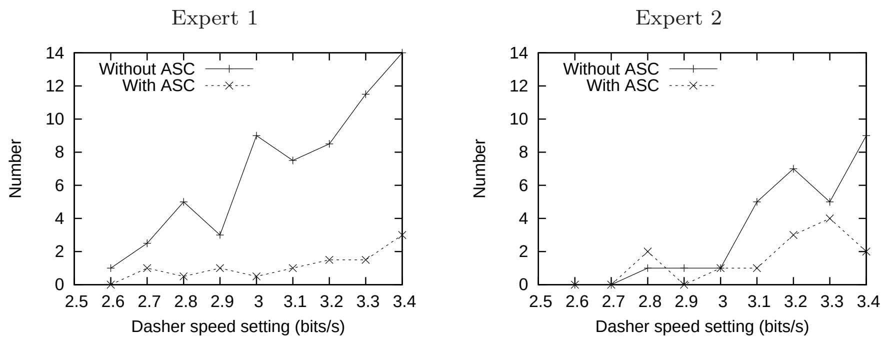


Fig. 8: Number of reverses used.

gesture significantly for expert 1. For expert 2 the improvement was less marked, but at the user's preferred speed (2.8 bps) the bits conveyed per gesture improved slightly on using the ASC, and the pressing rate decreased. Slightly fewer words are written per minute when expert 2 uses the ASC, whereas expert 1 writes more.

The ASC had a greater effect on the performance of expert 1 than of expert 2. A possible explanation is as follows: Firstly, expert 2 used the reverse fewer times than expert 1. The primary gain of the ASC is in cutting out reverses, so if a user never has to make a reverse, the ASC will actually be detrimental to his performance (through slowing the speed unnecessarily). Secondly, the ASC is based on the hypothesis that reverses are preceded by a 'small' Δt – for expert 2, this was only upheld (Figure 7) at lower speeds, and in this speed region, as Figure 6 shows, the ASC worked well, increasing the bits conveyed per gesture.

A possible modification that could be made is for the ASC to be triggered only when Dasher has observed that reverses in the past have been preceded by rapid button presses, thus separating these two cases and applying the appropriate behaviour.

The predictions of the theoretical model of Section 2.2 (Equation 8) are compared with the performances of the experts (without the ASC) in Figure 9. The model has a single free parameter for each user, the average timing error, μ, which was chosen by least squares, giving $\mu = 110\,\text{ms}$ for expert 1, and $\mu = 89.2\,\text{ms}$ for expert 2.

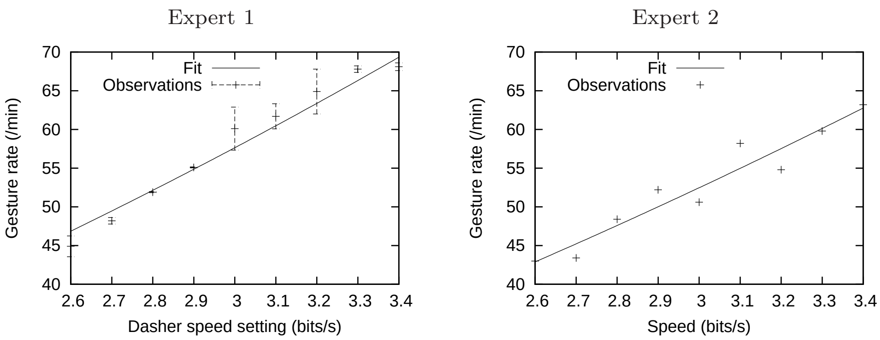
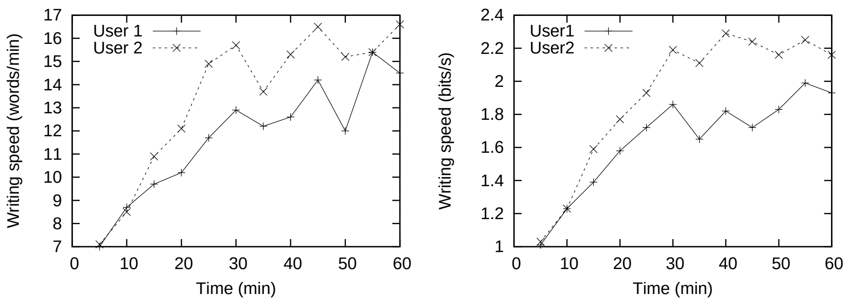


Fig. 9: Gestures per minute for two-button mode: Experimental observations and least squares fit using (8) Left: expert user 1, with $\mu = 110\,\text{ms}$, Right: expert user 2 with $\mu = 89.2\,\text{ms}$.



(a) (b)

Fig. 10: Experimental results for one-button mode. Results show (a) text entry rate and (b) information rate.

5. ONE-BUTTON MODE

5.1 Preliminary Experimental Results

One-button mode was tested on two subjects who were experts with two-button and normal Dasher but who had never used the one-button mode before. The trials consisted of 12 dictations of 5 minutes each. The apparatus was as in Section 3, except that only one button was used. After each dictation the subject had the option of increasing/decreasing speed by 0.1, 0.2 or 0.3 bps.

The results for the two subjects are shown in Figure 10. Word error rates are not shown as they were in most cases negligible (often 0% and always below 5%). The number of gestures made per minute at a given speed are shown in Figure 11. In cases where multiple trials were performed at the same speed, an average of the gestures made in each trial is taken.

5.2 Analysis

The maximum writing speed with one button was 16.6 words per minute using 61.4 gestures per minute. Unsurprisingly, this writing speed is lower than that obtained with two buttons by expert users.

In future trials, when expert subjects are available, we hope to do a direct comparison between the one- and two-button modes. We do however believe that these results indicate that the one-button mode shows promise as an efficient way of entering text with very limited hardware.

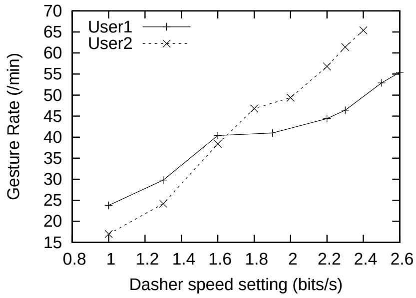


Fig. 11: Number of gestures made per minute as a function of the imposed zooming speed.

6. CONCLUSIONS AND FURTHER WORK

The button modes of Dasher allow gesture-efficient communication for people who are unable to make gestures at a high frequency, but are able to make them accurately. Using the two-button mode, novices could write at 6.9, 14.2 and 10.5 words per minute respectively after 1 hour of practice with virtually no errors. Experts using two-button Dasher wrote at 23.6 and 23.1 words per minute using 58.2 and 59.8 gestures per minute respectively. The speed control, which temporarily slows Dasher when it detects a high rate of button presses, helped one expert improve writing speed and allowed both experts to communicate more bits with each gesture. Preliminary results for the one-button mode indicate that a novice user can write at 16.6 words per minute after one hour's practice.

The two-button mode is now ready to test on users with disabilities in order to identify any issues which were not observed by the able-bodied users in these trials. While the initial results are promising, the one-button mode has not been as extensively tested, and must therefore undergo more extensive trials similar to those performed for the two-button mode.

REFERENCES

Bell, T. C., Cleary, J. G., and Witten, I. H. 1990. *Text compression.* Prentice Hall, Englewood Cliffs.

MacKay, D. J. C. 2003. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press. Available from `http://www.inference.phy.cam.ac.uk/mackay/itila/`.

MacKay, D. J. C., Ball, C. J., and Donegan, M. 2004. Efficient communication with one or two buttons. In *Proceedings of* Maximum Entropy and Bayesian Methods, R. Fischer, R. Preuss, and U. von Toussaint, Eds. AIP Conference Proceedings, vol. 735. American Institute of Physics, Melville, New York, 207–218.

Shorrock, T. H., MacKay, D. J. C., and Ball, C. J. 2005. Efficient communication by breathing. In *Machine Learning Workshop*, J. Winkler, M. Niranjan, and N. Lawrence, Eds. LNAI, vol. 3635. Springer, Berlin, 88–97.

Ward, D. J., Blackwell, A. F., and MacKay, D. J. C. 2002. Dasher – A data entry interface using continuous gestures and language models. *Human-Computer Interaction 17,* 2-3, 199–228.

Ward, D. J. and MacKay, D. J. C. 2002. Fast hands-free writing by gaze direction. *Nature 418,* 6900, 838.

Witten, I. H., Neal, R. M., and Cleary, J. G. 1987. Arithmetic coding for data compression. *Communications of the ACM 30,* 6, 520–540.