Efficient Communication Through The Timings of One Or Two Buttons

James Mead Phil Cowans
David MacKay

Cavendish Laboratory,
19 J J Thomson Avenue
University of Cambridge,
United Kingdom,
CB3 0HE
jmgm3@cam.ac.uk, pjc51@cam.ac.uk, mackay@mrao.cam.ac.uk

February 19, 2007

Abstract

Dasher is a communication system in which writing is a navigational process. Here we evaluate versions in which navigation is controlled using time critical 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. In both cases it is concluded that Dasher is a very gesture efficient communication system, ideal for users for whom every gesture is an effort.

1 An Introduction To Dasher

Dasher is a gesture efficient keyboard alternative based on Arithmetic Coding (for an account see [6] or [1]). Arithmetic coding is an optimal method for text compression using a language model. By turning arithmetic coding on its head, we obtain an optimal method for text generation. The Dasher approach to interface design decouples the issues of efficient bit-generation and efficient language-generation. Unlike in most interfaces, a Dasher user's gestures have no relationship to particular symbols in the language. Instead, they control navigation in a continuous space whose contents are laid out using a language model. To try Dasher out yourself (it's free) please visit *www.inference.phy.cam.ac.uk/dasher/*. In this paper we explore a new mode of Dasher that allows navigation through the use of two buttons and present preliminary results of another mode that allows navigation through the use of just one button. The first section of the paper (sections 2-5) describe analysis performed of two-button mode. Section 6 describes the one-button mode.

2 How Two-Button Dasher Works

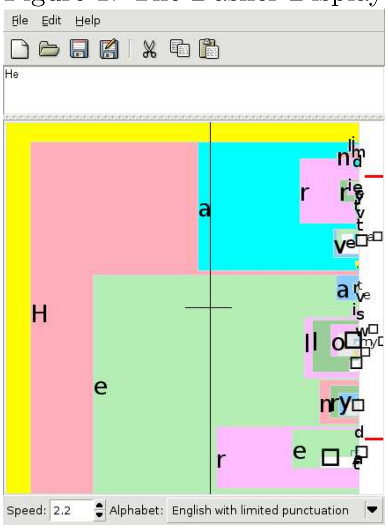
2.1 The Basic Idea

Imagine writing a piece of text by going into a library that contains all possible books and finding the book that contains exactly that text. In this way writing can be turned into a navigational task. What is written depends on where the user goes. In Dasher's idealized library, the 'books' are arranged alphabetically on one shelf.

In Dasher's two-button dynamic mode, navigation is achieved as follows. There are two 'fiducial' markers (horizontal red lines – they can be seen in figure 1) to the right of this display. The view zooms continually onto the central part of the display until the user presses a button, at which point the view shifts so as to centre the point of the display adjacent to the selected fiducial. The two buttons control which fiducial is selected; one is associated with each fiducial. To write a message, one first waits for the view to zoom until the desired location is adjacent to a fiducial, at which point the relevant button is pressed.

The writing process is made efficient by the use of a language model, which predicts the probability of the use of each letter within a given context and allocates shelf space accordingly. The language model used by Dasher is PPMD5. A picture of the Dasher interface while writing the word 'hello' is shown below in figure 1,

Figure 1: The Dasher Display

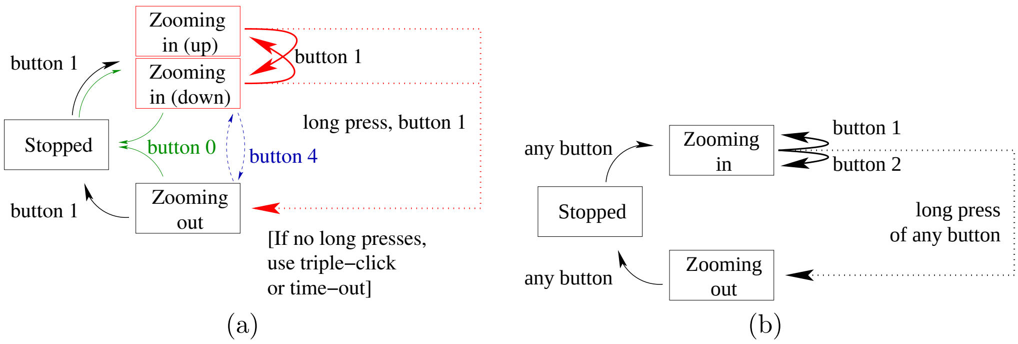


Both modes of button Dasher are aimed at people for whom every click is an effort – they are able to make precise clicks, but only at a low frequency. Precise clicks are 'rewarded', as they convey a lot of information. In this paper, an underlying theme is the ability of button Dasher to generate a large amount of information per gesture.

2.2 Error-Correction, Starting and Stopping

The zooming is started with a button press. A 'long hold', which in these experiments was 1 second long, will cause unzooming. This parameter is adjustable, however. For users who are unable to press for a long time, but are able to deliver rapid button presses, an additional method of reversing is to click a certain number of times within a certain time period (both the number of clicks and the time period within which they must be delivered can be set by the user). The interface is stopped by any button press whilst unzooming.

Figure 2: State diagrams for one-button mode (a) and two-button mode (b)



3 Theoretical Analysis of Two-Button Mode

τ is the time taken to enter one bit of information. The speed of Dasher, measured in bits per second (bps) is $\frac{1}{\tau}$. The efolding time, τ_e, is defined as $\frac{\tau}{\ln 2}$. Distances are measured in terms of screen coordinates; the centre point of the screen is at a vertical height 0, the top of the display at a vertical height 1, and the bottom at a vertical height -1.

To write a word, the user aims to click when the fiducial is adjacent to the desired point on the screen. However, there will be an error associated with this clicking action – the user clicks when the fiducial is a certain distance away from the desired location (where the distance is measured in terms of the display coordinates). If we denote the error immediately after the click as ϵ_0, then after a time, t, spent zooming , the vertical distance between the centre of the display and the destination, will be

$$\epsilon(t) = \epsilon_0 e^{\frac{t}{\tau_e}} \tag{1}$$

At time T the fiducial is aligned with the desired location. The fiducial position, measured in screen coordinates is ϕ.

$$\phi = \epsilon_0 e^{\frac{T}{\tau_e}} \tag{2}$$

The quantity $\frac{1}{T}$ is the clicking rate that must be sustained by the user to write accurately.

The user does not click exactly at the optimum time, T, but instead at a slightly different time, T_{click}. Define $t_{\text{error}} = T_{\text{click}} - T$ A further 'speed' can be defined as $\frac{\partial \epsilon}{\partial t} = \frac{\epsilon(t)}{\tau_e}$. This determines how fast the distance between the desired location and the screen centre grows with respect to the on-screen vertical coordinates. Taking $\epsilon(t) = \phi$, the distance of the fiducial from the desired location (the error) at the time of clicking T_{click}, is given by the timing error multiplied by the aforementioned speed,

$$\epsilon_0 = \frac{t_{\text{error}}\phi}{\tau_e} \tag{3}$$

We can now calculate an upper bound to the average clicking rate, denoted $\langle \Gamma \rangle = \left\langle \frac{1}{T} \right\rangle$, at a given speed. Using (2),

$$\langle \Gamma \rangle = \frac{1}{\tau_e \left\langle \ln\left(\frac{\phi}{\epsilon_0}\right) \right\rangle} \tag{4}$$

Using Jensen's Inequality, $\langle \ln\left(\frac{\phi}{\epsilon_0}\right) \rangle \geq \ln\left(\frac{\phi}{\langle \epsilon_0 \rangle}\right)$. From (3), $\langle \epsilon_0 \rangle = \frac{\phi \langle t_{\text{error}} \rangle}{\tau_e}$. Writing $\langle t_{\text{error}} \rangle = \mu$, the bounding inequality is,

$$\langle \Gamma \rangle \leq \frac{1}{\tau_e \ln\left(\frac{\tau_e}{\mu}\right)} \tag{5}$$

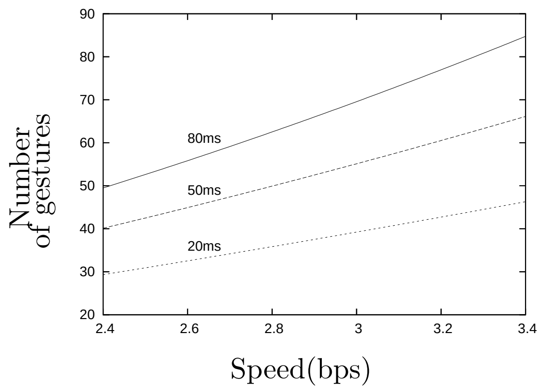


Figure 3: Predictions of (11) for different values of μ

The upper bound of (11) is plotted for a variety of μ in figure 2,

4 User Trials on Novice Subjects

The first set of user trials aimed to test how people improved in using two-button mode over a period of 1 hour.

4.1 Experimental Procedure

1. Subjects

   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.

2. Task

   The task was to enter text dictated from Jane Austen's *Emma*. 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.

3. Apparatus

   (a) **Platform**: Pentium IV 2.8GHz running Linux 2.8.3

   (b) **Dasher Display**: 700x491 pixels, font size 12

   (c) **Input Device** Two buttons, using a Don Johnston switch interface. The trialists could zoom out ('reverse') by holding either one of these buttons down for half a second. The subjects used one hand to operate switches.

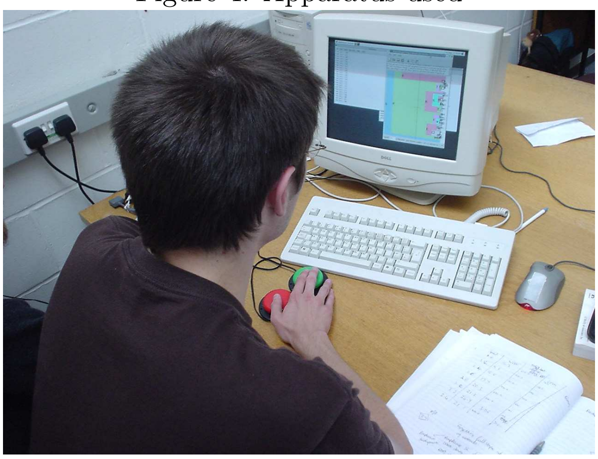
   (d) **Dictation** The passages were recorded as a series of audio files and cycled through by a demonstrator

   (e) **Language Model** PPMD5 trained on a corpus from Emma, excluding dictation passages.

   (f) **Dasher Settings** 1 second long hold to reverse, fiducials at screen coordinates $\pm$ 0.8.

   The apparatus is pictured in figure 3,

Figure 4: Apparatus used



4. Procedure

The protocol is similar to that in [5]. The subjects used Dasher in 6 sessions, with two 5 minute periods of writing within each session. Within these 5 minute periods the trialists were read a dictation from Jane Austen's *Emma*, and entered the text they heard using the Dasher interface. The dictation passages were stored as audio files on a computer and cycled through by the dictator as required. Before the first session, trial participants 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 1.0bps. Before each dictation, the trialists 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 user was given the option of increasing/decreasing speed by 0.1, 0.2 or 0.3bps. Between each 5 minute writing spell the users 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.

4.2 Results

Figure 4 shows the results for the novice subjects. The number of words written is defined to be the number of characters divided by five. The most common form of error was the omission of full stops, which users found difficult to locate, although with practice all users learnt to write very accurately, with all error fractions below 5% after 1 hour.

5 Expert Users With and Without Automatic Speed Control

Expert users, both with over 5 hours of experience of two-button mode, were tested. An automatic speed control (ASC) was developed. Rapid clicking is a sign of distress and inefficiency (failing to communicate the maximum number of bits with each gesture). The ASC slows the speed whenever Dasher records a Δt below some adjustable factor multiplied by the median Δt. By slowing the speed, the user has time to make the next click accurately and recover. Without the slow down, the rapid clicking produces a 'cascade' of inaccurate clicks which will lead to failure.

5.1 Procedure

An expert underwent four 45 minute sessions; two with the ASC and two without. Another 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 0.1bps from 2.6 bps to 3.4 bps. The ASC slowed the speed by 90% whenever a Δt below 0.5 of the median, or a Δt below 0.3s was detected, whichever is greater. The latter condition is imposed because 0.3s was the minimum time between clicks the buttons allowed. Dasher would

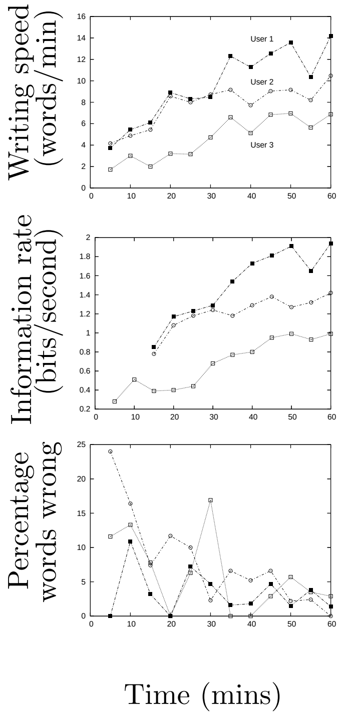


Figure 5: Data from novice subjects

then accelerate back to its original speed over 1 second. Dasher measured the $\Delta t's$ as it ran, and formed the distribution from which the median was drawn from the first 100 clicks.

5.2 Results

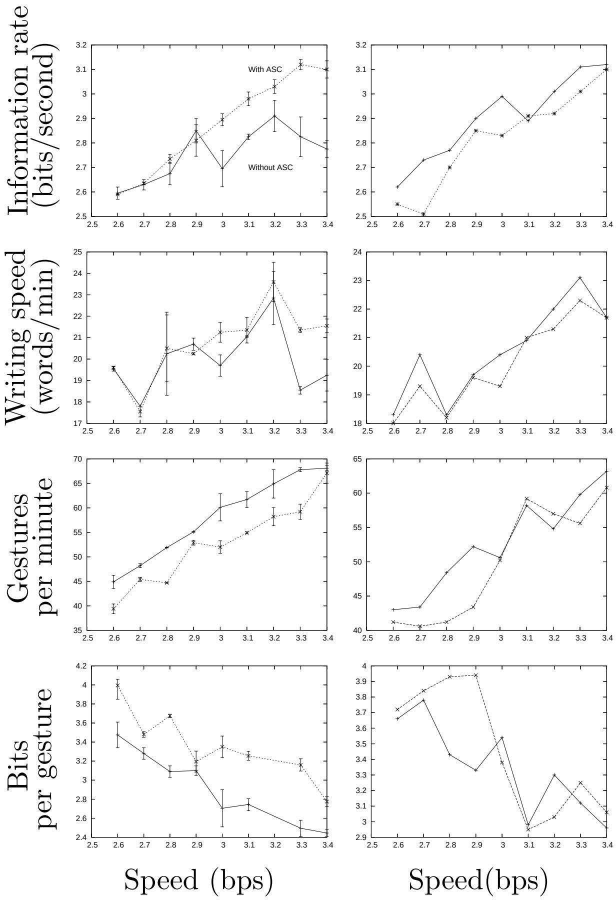
Writing data is plotted in figure 5. For expert 1, the averages from the two relevant sessions are plotted, with the errors taken as half the range divided by $\sqrt{2}$. The error percentage is not plotted, as for both experts this was negiligible.

The fraction of reverses which were preceded by a Δt of less than half the median (without the ASC) is plotted in figure 6. The average number of reverses used at each speed is plotted in figure 7.

5.3 Analysis

The useful region of these graphs lies before the 'optimum' speed, as even writing at this speed can feel uncomfortably fast, and it is likely that most users will choose to operate at speeds lower than this. Expert 1 found the most comfortable speed to be at 3.0 bps, and expert 2 at 2.8 bps. These are below the speeds at which the maximum writing speed was attained.

The ASC has almost eradicated the use of the reverse (figure 7). This is an important feature of the ASC if two-button mode is to be used by real disabled users, at whom it is aimed – for many, the long hold could be a difficult gesture



(a) Expert 1 (b) Expert 2

Figure 6: Expert data

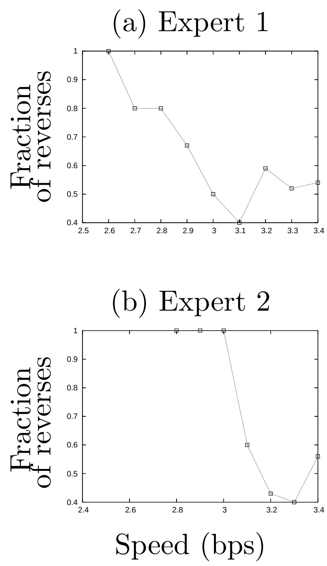


Figure 7: Fraction of reverses without the ASC preceded by a Δt less than half of the median.

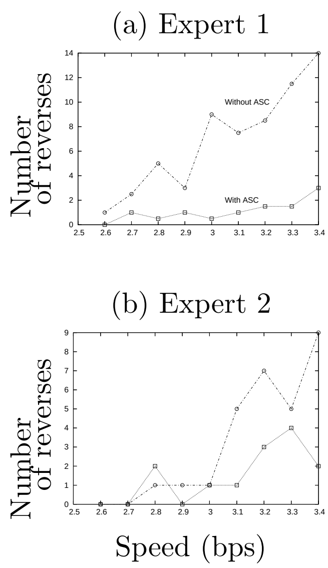


Figure 8: Number of reverses used

to make, and so any system that cuts down the number of intensive long-hold gestures will be highly desirable.

From figure 5, the ASC improves the information rate and bits conveyed per gesture significantly for expert 1. For expert 2 the improvement was less marked, but in the region of interest below the optimum the bits conveyed per gesture improved slightly on using the ASC. Fewer words are written per minute when expert 2 uses the ASC, whereas expert 1 is able to write more.

The ASC had more effect on the performance of expert 1 than of expert 2. The explanation is simple. 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 unecessarily). Secondly, the ASC is based on the hypothesis that reverses are preceded by a 'small' Δt – for expert 2, this was only upheld (figure 6) at lower speeds, and in this speed region, on observation of figure 5, the ASC worked well, increasing the bits conveyed per gesture.

In future versions a possible modification that could be made is that the ASC is only triggered when Dasher detects the fraction of reverses preceded by a 'small' Δt above 0.6 – this way we can be sure that the user is both using the reverse, and that reverses are preceded by rapid clicking, which are our two key requirements for the ASC to work well.

Button-Dasher can be compared with Morse. Using character probabilities from *Emma* and the number of gestures required to write each character in Morse, the mean number of gestures required per character is 2.1925, which allows the values in the table below to be calculated. In the table below the comparison is made. It is clear that Button-Dasher is very gesture-efficient compared to Morse; Morse requires *four* times as many gestures, even without capitalisation.

	Optimum speed (bps)	**Average Gestures per min**	
		Dasher	**Morse**
Expert 1 without ASC	3.2	64.9	250
Expert 2 without ASC	3.3	59.8	250
Expert 1 with ASC	3.2	58.2	260
Expert 2 with ASC	3.3	55.6	240

The predictions of equation (11) are compared with the performances of the experts (without the ASC) in figure 8. $\mu = 55ms$ was used for expert 1, and $\mu = 45ms$ was used for expert 2. It has been assumed that μ remains constant over the speed range.

6 One-Button Mode

In [2], a theoretical model of a single-switch user is presented. As in [2], it's assumed that the user controls only the times of presses, not the times of releases and that the user cannot distinguish between short and long presses.

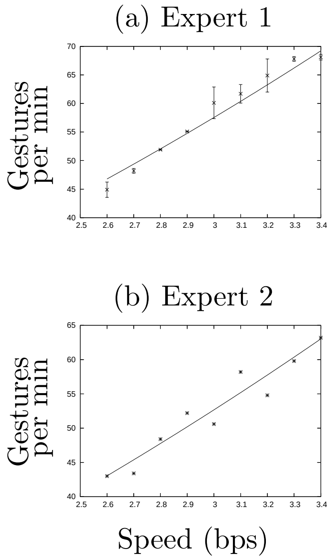


Figure 9: Gestures per minute: theoretical predictions and experimental observations

6.1 Theoretical Analysis

The user is modelled with two parameters: a timing accuracy g, and a recovery time S. The user clicks within a time $\pm g/2$ around each intended click time, and then requires a delay of duration S before she can click again.

As in [2], time is divided into boxes of duration g. The user's actions are a sequence of choices between two actions, '0' (doing nothing, which takes time g), and '1' (pressing the switch, which takes time S). As soon as an action has been completed, the user is free to make another choice.

The Capacity is C, where

$$p_1 = 2^{-CS} \tag{6}$$

$$p_0 = 2^{-Cg} \tag{7}$$

and

$$p_1 + p_0 = 1. \tag{8}$$

The information rates during a '1' and '0' can be calculated. The information content conveyed by choosing 1 is $\log_2 \frac{1}{p_1}$, and it occupies a duration S. So the information rate is $\frac{\log_2 \frac{1}{p_1}}{S}$. Performing a similar calculation for a '0', the information rate is $\frac{\log_2 \frac{1}{p_0}}{g}$; both are equal to C bits per unit time. This indicates that Dasher should zoom at a steady rate. This observation motivates a very simple idea. Using the same screen coordinates as before, the Dasher screen is split into two sections of length p_0 and p_1. An event '1' zooms Dasher into the section of length p_0 and an event '0' zooms Dasher into the section of length p_1. After every button press the positions of the two sections are reversed.

6.2 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 left unchanged from the previous sections, except there was only one button to press. After each dictation the subject had the option of increasing/decreasing speed by 0.1, 0.2, 0.3bps.

The results for the two subjects are in figure 9; it does not include data on the percentage of words with errors and this was often 0% and always below 5% – this is because both users were already very familiar with the Dasher concept, and so many of the most common mistakes in the previous novice trials (such as missing full stops) were not made by them.

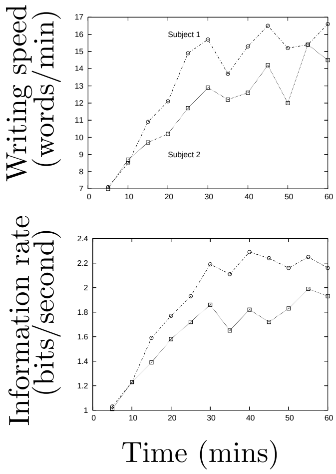


Figure 10: Data from subjects

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.

6.3 Analysis

The maximum writing speed was 16.6 wpm using 61.4 gpm. The mode also remains faithful to the defining feature of the button Dasher modes – the ability to communicate information in a gesture-efficient manner, particularly at lower speeds. The one-button subjects were not experts, and with practice the number of gestures required to write a given quantity of information should fall. In future trials, when expert subjects are available, we hope to do a direct comparison between the one- and two-button modes.

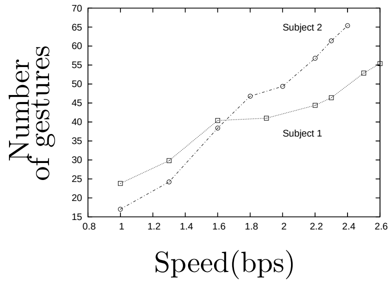


Figure 11: Number of gestures made per minute at different speeds

7 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 slowed Dasher when it detected a high clicking rate, 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. [James , please confirm. this addition 'after one hour's practice'.]

The two-button mode is now ready to test on real, disabled users. Throughout these trials, able-bodied subjects used Dasher, but their behaviour may be different from a disabled subject who is physically restricted to a certain gesture frequency. The one-button mode has not been as extensively tested. Tests on expert subjects are required, and after that, tests on disabled users.

References

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

[2] David J. C. MacKay, Chris J. Ball, and Mick Donegan. Efficient communication with one or two buttons. In Rainer Fischer, Roland Preuss, and Udo von Toussaint, editors, *Proceedings of* Maximum Entropy and Bayesian Methods, volume 735 of *AIP Conference Proceedings*, pages 207–218, Melville, New York, 2004. American Institute of Physics.

[3] Thomas H. Shorrock, David J. C. MacKay, and Chris J. Ball. Efficient communication by breathing. In M. Niranjan J. Winkler and N. Lawrence, editors, *Machine Learning Workshop*, volume 3635 of *LNAI*, pages 88–97, Berlin, 2005. Springer. (in press).

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

[5] D. J. Ward and D. J. C. MacKay. Fast hands-free writing by gaze direction. *Nature*, 418(6900):838, 2002.

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