

Dasher – an efficient keyboard alternative

David MacKay

Keyboards are large, inefficient text-entry systems. They are inefficient for two reasons.

1. Keyboards do not take advantage of the predictability of normal writing. Whereas a choice of one key from a keyboard of 64 characters might convey 6 bits of information (since $2^6 = 64$), the information content of English is only about 1 bit per character. So keyboards are immediately inefficient by a factor of six.
2. Keyboards throw away the user's ability to make fine, continuous motor gestures. One finger, for example, is an analog device capable of generating many bits per second of precise pointing information; but a keyboard reduces the action of a finger to a single bit: up or down.

Dasher is an invention intended to rectify both these inefficiencies, by coupling a human's natural pointing capabilities directly to an interface that models the predictability of the text. *Dasher* is a piece of software for text-entry, driven by continuous one- or two-dimensional gestures, delivered, for example, via a mouse, touch screen, or eyetracker; the user writes by steering through a continuously expanding two-dimensional world containing alternative continuations of the text, arranged alphabetically. *Dasher* uses a language model to predict which letters might come next and makes those letters easier to write. The language model can be trained on example documents in almost any language, and adapts to the user's language as she writes. *Dasher* is free software.

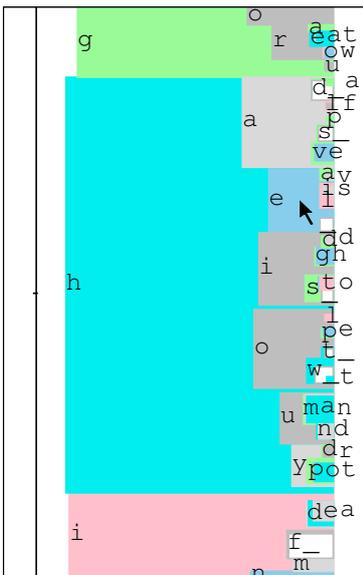


Figure 1. A screenshot of *Dasher* when the user starts writing `hello`. The shelf of the alphabetical ‘library’ is displayed vertically. The space character, ‘_’, is included in the alphabet after `z`. Here, the user has zoomed in on the portion of the shelf containing messages beginning with `g`, `h`, and `i`. Following the letter `h`, the language model makes the letters `a`, `e`, `i`, `o`, `u`, and `y` easier to write by giving them more space. Common words such as `had` and `have` are visible. The pointer's vertical coordinate controls the point that is zoomed in on, and its horizontal coordinate controls the rate of zooming; looking to the left makes the view zoom out, allowing the correction of recent errors.

HOW DASHER WORKS

Imagine writing a piece of text by going into the 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 is determined by where the user goes. In Dasher's idealized library, the 'books' are arranged alphabetically on one enormous shelf. When the user points at a part of the shelf, the view zooms in continuously on that part of the shelf. To write a message that begins 'hello', one first steers towards the section of the shelf marked **h**, where all the books beginning with **h** are found. Within this section are sections for books beginning **ha**, **hb**, **hc**, etc.; one enters the **he** section, then the **hel** section within it, and so forth.

To make the writing process efficient we use a language model, which predicts the probability of each letter's occurring in a given context, to allocate the shelf-space for each letter of the alphabet, as illustrated in figure 1. When the language model's predictions are accurate, many successive characters can be selected by a single gesture. With Dasher, it is easy to spell correctly and hard to make spelling mistakes.



Figure 2. Dasher can be driven by eyetracker or by pointing on a touchscreen, as well as with a regular mouse.

WRITING SPEEDS WITH DASHER

The user steers using any convenient pointing system. The simplest is an ordinary mouse attached to an ordinary PC. Using a mouse, typical novice users reach a writing speed of 25 words per minute after 60 minutes of practice, and expert users can write at 35 words per minute. Dasher can also be driven more directly using a computer with a touchscreen; it works nicely on a Pocket PC. Dasher does not need great pointing precision.

For users who cannot point using a conventional mouse or touchscreen there are two ways in which Dasher can be used hands-free with a PC. The cheapest solution is a head mouse: a reflective dot is attached to the user's head (or whatever piece of anatomy they wish to move) and a small camera tracks the dot to control the mouse. For severely paralyzed people, the direction of gaze can be tracked using an eyetracker. After 60 minutes' practice, novice users can drive Dasher using an eyetracker at a speed of about 15 words per minute; expert users can write at 25 words per minute. Not only is this speed much faster than alternative hands-free systems such as on-screen keyboards; Dasher users make far fewer spelling mistakes.

Furthermore, whereas staring at on-screen buttons is exhausting, navigating through the Dasher landscape is a natural activity for the eyes, comparable to driving a car.

WHO IS DASHER FOR?

It would be nice to create a new writing system superior in all ways to a keyboard, but ten-finger typists should not throw away their keyboards yet. Dasher is not as fast as ten-finger typing. But it is useful to a large community of users, both able-bodied and disabled.

Anyone who cannot use a regular keyboard will find Dasher useful – for example those who suffer from repetitive strain injury. Dasher can be driven using an ordinary mouse and does not require button-clicking. It can also be driven by numerous unconventional mice; we find the Smart-Nav head mouse is an especially versatile solution.

Dasher can also be used by severely disabled people. As I mentioned above, anyone with normal eyesight can use Dasher with an eyetracker to communicate at up to 25 words per minute. We are also developing a breath-controlled version of Dasher, with lung volume directly controlling the vertical coordinate of the mouse. In this situation, the user conveys only a one-dimensional pointing signal, whereas regular Dasher uses a second horizontal pointing dimension to control the speed of zooming-in or zooming-out. We have made a one-dimensional-pointing version of Dasher in which extreme pointing, up or down, causes the display to zoom out, and the intermediate range of pointing gestures give the normal zooming-in behaviour. We find this one-dimensional version just as easy and fast to use as two-dimensional Dasher.

Finally, Dasher is perfect for miniature computers (including mobile phones) and for tablet PCs that do not have full-size keyboards. Compared with hand-writing-based systems, Dasher has a much smaller error-rate. And compared with miniature on-screen keyboards, Dasher is faster (after a little practice) and requires less precise pointing by the user. If a palmtop computer had a tilt sensor in it, then Dasher, in its one-dimensional mode, could also be used one-handed.

LINGUISTICS AND DASHER

The language model inside the current version of Dasher is embarrassingly crude. The model simply records the frequencies, in the training text, of all letters, all pairs of letters, all trigrams, and so forth, up to sextuplets, and merges these statistics to make predictions in any given context. So Dasher knows nothing of the concepts of words, dictionaries, or grammar. As Shannon established in 1948, most of the predictability of English is captured in its letter-level statistics. And Dasher certainly behaves as if it knows not only words but also whole phrases.

One advantage of this crude approach is that it allows a single piece of software to work instantly in multiple languages and multiple fonts. Dasher version 1.6 works in Japanese and Dasher version 3.0 works in French, German, Spanish, Polish, and Portuguese. To switch language, one simply loads up a training file containing natural text in the chosen language. And because the language model learns all the time, it can pick up the user's personal style and turns of phrase. A new language can be added to Dasher's repertoire by editing an XML file that specifies the letters of the alphabet. Dasher version 3 works in Unicode, so the International Phonetic Alphabet could easily be implemented.

In a domain where strict grammatical rules are obeyed, for example in symbolic languages, it might be worth replacing the simple "PPM" language model (prediction by partial match)

by a more sophisticated model that captures the grammar.

Is Dasher useful for language acquisition, for learning to spell, or for other educational purposes? Formal studies have yet to be made. But informal evidence of the utility of Dasher to everyday linguistic development comes from a user in New Zealand, who describes the response of three children to Dasher:

While playing with Dasher was making me a little seasick, not so my 11 year old son. He's a computer wizz and was asking how to speed it up in no time. He's played every game imaginable, and written even more of his own, yet Dasher held his attention for several hours, days in a row. That's remarkable. Furthermore, we had a couple of kids here yesterday, at the opposite end of the spectrum: they have no home computer and few at their school (almost no access to them anyway). But they too took to Dasher and were fascinated. The three argued over spelling and sentence construction - and even got the dictionary out at one point to confirm/deny.

That's the long story, the short one is, while I can't pin down the exact appeal, I strongly suspect Dasher has a place in education as well as special needs. In that region where those two meet, it could be a very potent educational tool.

THE FUTURE OF THE DASHER PROJECT

Dasher was created by David MacKay and David Ward in the Physics department of the University of Cambridge. The project is supported by the Gatsby charitable foundation. Dasher currently works on PCs running Windows or GNU/Linux, on MacOS-X, and on Pocket PCs.

Most users find Dasher is quick to learn, just like a video game – ‘attack of the killer alphabets’, it’s been called. We encourage you to try it out (it’s free!) and send us your feedback.

www.inference.phy.cam.ac.uk/dasher/

Draft 1.2

Written for *Glott International*, a web based linguistic journal.

© David MacKay July 10, 2003.