Breaking the Laws of Action in the User Interface

Per-Ola Kristensson
Department of Computer and Information Science
Linköpings universitet
perkr@ida.liu.se

Abstract

Fitts' law, Steering law and Law of crossing, collectively known as the laws of action, model the speed-accuracy trade-offs in common HCI tasks. These laws impose a certain speed ceiling on precise actions in a user interface. My hypothesis is that for some interfaces, the constraints of these laws can be relaxed by using context information of the task. To support this thesis, I present two systems I have developed for pen-based text input on stylus keyboards. These systems break either Fitts' law or the Law of crossing by taking advantage of high-resolution information from the pen, and the fact that words can be seen as patterns traced on the keyboard. Using these systems users can potentially gain higher text entry speed than on a regular stylus keyboard that is limited by the laws of action. I conclude by discussing planned future research, primarily improved visual feedback and empirical evaluation.

Categories & Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *input devices and strategies, theory and methods*; I.5.5 [**Pattern Recognition**]: Implementation – *interactive systems*

General Terms: Human Factors, Theory

Keywords: Laws of action, stylus keyboard, Fitts' law, shorthand, pattern recognition

INTRODUCTION

One of the most well known human performance laws is probably Fitts' law [1], which models the predicted mean time T it takes to point at a target of width W over distance D as:

$$T = a + b\log_2\left(\frac{D+W}{W}\right) \qquad (1)$$

where a and b are regression coefficients. This empirical law has found many uses in HCI. Although Fitts' law is very useful for modeling pointing actions, it cannot model more complicated actions such as steering a pointer through a tunnel (e.g. maneuvering the mouse pointer through a menu structure) or crossing a target. As a response, the Steering law and Law of crossing were derived, and experimentally validated. See Zhai et al. [5] for a summary of this line of work.

Fitts' law, the Steering law and the Law of crossing, collectively known as the "laws of action" are laws that model the average time it takes for a user to point, steer or cross in a visually-guided manner in a user interface. As such, these laws limit how fast a user can perform these actions. If a user tries to perform an action faster than the predicted mean time for the action in a regular user interface, the user will tend to miss the target or steer outside the tunnel.

For some user interfaces it is possible to do better than what the laws of action predict. It is interesting to note that Fitts' law was inspired and derived from Shannon's information theory. One interpretation of Fitts' law is that if the amount of information as specified by Fitts' index of difficulty has to be expressed in the movement of the performer, one has to use the amount of time Fitts' law predicts. My central hypothesis is that if information constraint (context) beyond the individual movement is utilized, these laws of action can be "broken", in the sense that the target width or tunnel constraint can be artificially magnified. Such a thesis is quite plausible in general. The challenge lies in developing specific techniques and systems to enable the users to go beyond what the laws of action require. The interface I have been concentrating on is the stylus keyboard, or "graphical" keyboard.

BREAKING FITTS' LAW

A stylus keyboard is a very sensitive interface. If the user is one pixel outside the desired target key of the keyboard, an erroneous key press will be reported. Tapping on a regular stylus keyboard is also a visually-guided closed-loop action that is limited in time performance by Fitts' law.

However we can artificially break the W constraint in Equation 1 by making two key observations. First, not all possible letter combinations on a keyboard are legitimate words in the language; hence we can limit possible input strings to a lexicon of words. Second, both the center positions of the keys that comprise a word in the lexicon, and the user's sequence of stylus hit points, form high-resolution geometrical patterns. These patterns can be compared using pattern recognition methods. The best correspondence between a user's pattern of stylus hit points and the patterns of words that make up the lexicon can then be searched. If a close match is found, the word from the lexicon can be outputted instead of a verbatim translation of the keys the user actually pressed. For example in Figure 1 the user tapped on the keys *r*, *j*, *n* and *r* in sequence. Despite missing all the keys of the intended word the, one can see from the figure (remembering that the hit points are considered in sequence in the comparison) that the tap-

CHI 2005, April 2–7, 2004, Portland, Oregon, USA.
ACM 1-59593-002-7/05/0004.

ping pattern is close to the geometrical pattern of the keys comprising *the*. Using pattern recognition the system can still recognize this input pattern as *the*. This means that the relevant letter keys can be seen as being artificially magnified. We call such a system an Elastic Stylus Keyboard (ESK). See Kristensson and Zhai [2] for the details on the algorithms and the evaluations.

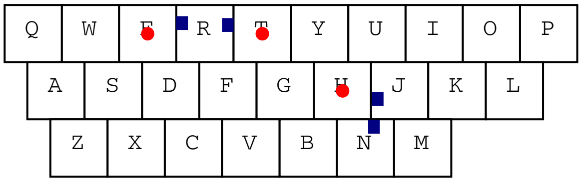

Figure 1. Circles shows ideal stylus hit positions. Rectangles indicate a user's actual stylus hit points.

SHORHAND WRITING ON STYLUS KEYBOARD

The Law of crossing can be broken in a similar fashion. By considering the letter keys comprising a word as a *trajectory pattern* as opposed to a sequence of stylus hit points, one can use a multi-channel architecture [3] to recognize literally thousands of word patterns, called *sokgraphs* – ***s****horthand* ***o****n* ***k****eyboard as* ***graphs*** (see Figure 2 for an example). The basic idea is that a novice user starts off by tracing the word pattern (that is, serially crossing the letter keys comprising the word, in the case of Figure 2 this would mean crossing the keys *t*, *h* and *e* in order). Over time the pattern builds up in the user's memory and the user can go from the closed-loop action of crossing the keys to the open-loop action of quickly flicking a shorthand gesture that is matched against the ideal sokgraph using pattern recognition methods. We call this method SHARK – ***Sh****orthand* ***A****ided* ***R****apid* ***K****eyboarding* [4]. Since the system uses pattern recognition, sokgraphs are partially scale and translation invariant. If recognition was not invariant to these features, the user would have to cross each key comprising a word. In this regard, SHARK breaks the Law of crossing.

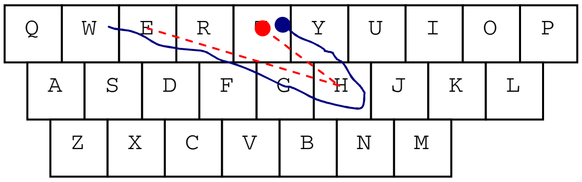

Figure 2. Solid trace shows a user writing the sokgraph for "the" open-loop. The dashed trace shows the ideal sokgraph.

CONCLUSIONS AND FUTURE WORK

The laws of action are important tools to design user interfaces that are as fast and accurate as possible. In this paper I have presented systems that break these laws, and rely on pattern recognition and the redundancies in the languages to enable users to enter text faster and more comfortably using a pen. Future work includes primarily four tasks: evaluation, improvements in the pattern recognition algorithms, better visual feedback, and transferring the technology to other HCI applications.

Evaluation

There are still important work left in evaluating both SHARK and ESK. I still do not have complete knowledge on the performance limit or learning curve of either system. Another interesting task is to compare these systems against each other: are there properties that make one system or another preferable in some context? The ESK for example has additional information (the stylus hit point that resides in proximity of a desired key) that is not easily derived from a gesture in SHARK. On the other hand, in SHARK the user can flick a word in one gesture, while a user of ESK has to explicitly tap each desired letter key. The trade-offs between these actions (when relaxing the laws of action) are unknown.

Pattern Recognition

Although both systems work satisfactory and SHARK is polished to the degree of being released on IBM alphaWorks (http://www.alphaworks.ibm.com/tech/sharktext/), better algorithms can potentially allow higher flexibility for expert users who quickly flick the gestures open-loop.

Visual Feedback

Both systems use pattern recognition to detect the user's input. In SHARK I have experimented with the use of morphing to gradually transform the user's gesture into the ideal recognized sokgraph. However, that technique only related the user's gesture against the recognized sokgraph. A richer visualization would reveal the "freedom of movement" in the interface, relating to *all* close sokgraphs. Such information would allow users to see how "sloppy" a particular gesture can be produced.

Breaking the Laws of Action in Other Domains

Last, I am also investigating how the techniques outlined in this paper can be used to improve other HCI interfaces that have a spatial representation and where high accuracy is important, such as finger-operated touch-pads.

ACKNOWLEDGMENTS

I would like to thank my Ph.D. advisor Shumin Zhai.

REFERENCES

1. Fitts, P.M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology,* 47, 1954, 381-391.
2. Kristensson, P-O. and Zhai, S. Relaxing Stylus Typing Precision by Geometric Pattern Matching. *Proc. IUI 2005,* 151-158, ACM Press.
3. Kristensson, P-O. and Zhai, S. SHARK^2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. *Proc. UIST 2004,* 43-52, ACM Press.
4. Zhai, S. and Kristensson, P-O. Shorthand Writing on Stylus Keyboard. *Proc. CHI 2003,* 97-104, ACM Press.
5. Zhai, S., Accot, J. and Woltjer, R. Human Action Laws in Electronic Virtual Worlds. *Presence*, 13(2), 2004, 113-127.