Evaluation of Multimodal Input for Entering Mathematical Equations on the Computer

Lisa Anthony, Jie Yang, Kenneth R. Koedinger
Human Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{lanthony, yang+, koedinger}@cs.cmu.edu

ABSTRACT

Current standard interfaces for entering mathematical equations on computers are arguably limited and cumbersome. Mathematics notations have evolved to aid visual thinking and yet text-based interfaces relying on keyboard-and-mouse input do not take advantage of the natural two-dimensional aspects of math. Due to its similarities to paper-based mathematics, pen-based handwriting input may be faster, more efficient, and more preferable for entering mathematics on computers. This paper presents an *empirical study* that tests this hypothesis. We also explored a multimodal input method combining handwriting and speech because we hypothesize that it may enhance computer recognition and aid user cognition. Novice users were indeed faster, more efficient and enjoyed the handwriting modality more than a standard keyboard-and-mouse mathematics interface, especially as equation length and complexity increased. The multimodal handwriting-plus-speech method was faster and better liked than the keyboard-and-mouse method and was not much worse than handwriting alone.

Author Keywords

Multimodal interfaces, mathematics interfaces, equation entry and editing, evaluation.

ACM Classification Keywords

H.5.2 [User interfaces]: *evaluation/methodology*, *input devices and strategies*, *user-centered design*, *voice I/O*.

INTRODUCTION

Current interfaces for entering equations are largely limited to keyboard- and mouse-driven windows-icons-menus-pointing (WIMP) interfaces. Advanced input consists of mark-up languages such as LaTeX and programming languages such as Mathematica and Maple. Both methods require learning a new language and syntax, and can be difficult for novices to grasp and slow for experts to use.

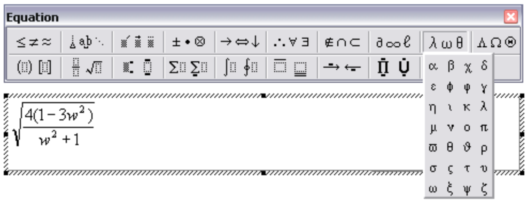
Paper-based mathematical notations are designed to represent and aid mathematical thinking and visualization. It is therefore natural and convenient for users to communicate with computers in the same way [2]. Because pen-based input can use traditional paper-based notations, it may be better with respect to speed, efficiency and user satisfaction for entering mathematics on computers. Several systems exist for handwriting-based input both online and offline [*e.g.*, 7], but are not widely available to most novices. In addition, no rigorous studies have been done to demonstrate the effectiveness of such interfaces over more available keyboard-and-mouse interactions. This paper presents an empirical study to do so that motivates further investigation of multimodal interfaces for mathematics. The study focuses on the input experience decoupled from the problems inherent in automatic recognition technologies. Users were not given feedback about how the computer may or may not recognize their input.

Although we expect these techniques to benefit general mathematical software package interfaces, an important application is for intelligent tutoring systems for math. Students using these tutors are novice mathematics learners and therefore are not likely to have used more complex tools for entering math on computers. A key pedagogical motivation for this is the instructional principle to minimize working memory or cognitive load (*e.g.*, [8,1]). Simplifying the interface to rely solely on the "language of mathematics" that students should be learning may help alleviate cognitive overload and could improve learning.

MOTIVATION

Given prior studies where typing was found to be faster than handwriting [*e.g.*, 3], one might ask why handwriting would ever be used instead of typing in mathematics. One reason is that mathematics often appears in higher-dimensional layouts, enabling the representation of both superscripts and subscripts, for instance, but most computer interfaces are optimized for entering linear text [7]. In addition, template-based interfaces that do allow higher-dimensional representations, such as Microsoft's Equation Editor [5], require the user to construct expressions in a top-down manner, making later structural changes difficult [7]. Furthermore, studies favoring typing over handwriting speed may not apply to equation entry because they have focused on entering paragraphs of English text. Standard keyboards do not allow users to easily type complex

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



$$\sqrt{\frac{4(1-3w^2)}{w^2+1}}$$

Figure 1. Microsoft's Equation Editor toolbar provides menus to allow users to enter special symbols, fractions, exponents, etc.

mathematical expressions such as fractions, exponents or special symbols like ∑ and √. It is possible that for simple linear equations, the keyboard may be faster. However, for longer, more complex equations, we hypothesized that handwriting would be faster and more efficient than typing. Although some systems that can recognize handwritten equations have reported evaluations [*e.g.*, 7], none of them have reported an evaluation of the handwriting modality de-coupled from technological limitations of their system with respect to recognition accuracy and correction of errors.

MATH INPUT STUDY

In our study, users were asked to enter mathematical equations of varying complexity using four different modalities: (1) traditional keyboard-and-mouse (**KB**) using Microsoft Equation Editor [5], (2) pen-based handwriting entry (**HW**), (3) speech entry (**SP**), and (4) handwriting-plus-speech (**HWSP**). Microsoft Equation Editor was chosen as a representative tool for novice users because it is in wide use and is a prime example of a WIMP interface. There was no automatic handwriting or speech recognition in our study; users simply input the equations and did not get feedback about computer recognition of their input.

Pairing handwriting and speaking may not immediately seem like a natural choice. We explored a multimodal input method combining handwriting and speech because we hypothesize such a combination of inputs might enhance computer-based recognition of equations [6] and could aid user cognition. Research has shown that people speak in an "inner voice" (subvocalization) while reading or writing [4]. We saw several users during the sessions who, in the speaking-only condition, wrote in the air with their hands while speaking the equation out loud. Exploring the pairing of these two modalities is important to supporting user cognition during handwriting input on the computer.

Participants

Forty-eight paid participants (27 male, 21 female), graduate or undergraduate students at our university, answered an ad to participate in this study. All participants were fluent English speakers with unaccented speech. We saw no effects of age or ethnicity in our analyses. Most (33) had no experience with Microsoft Equation Editor before the study. Of those who knew of it or had used it, only 2 classified themselves as knowing it "very well."

Procedure

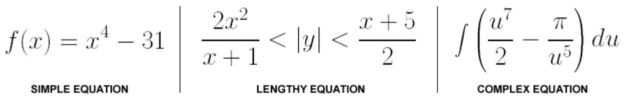
The experiment was a within-subjects design wherein participants came to the lab for a 45-minute session and entered mathematical equations on a Tablet PC in four different conditions. There was a list of 36 equations (9 per condition) which remained constant for all participants; order of presenting each condition was counterbalanced across all possible orderings. Participants also answered a questionnaire before the session in which they rated their pre-existing preferences for each condition. Before performing each condition, participants were instructed as to how to enter equations in that condition. For instance, in the HW condition, the experimenter explained that the stylus could be used like a regular pen on paper. The experimenter did not tell the participants in what format to write the math, or how to find certain symbols or express things. Participants were given a 5-minute "practice" period before the KB condition to familiarize themselves with the Microsoft Equation Editor toolbar (Figure 1). During this time, they explored on their own with no feedback or input from the experimenter. Although there was no exploratory period for the other three conditions, the first two equations in each condition were considered practice and were not included in our analyses. When participants finished all four conditions, they answered a questionnaire again rating their preferences for entering equations in each condition.

Stimuli Design

The experimental stimuli (36 equations) were designed with two components in mind: (1) the number of characters in the equation, and (2) the number of "complex" symbols appearing in the equation such as fractions, exponents, special symbols, and so on. Figure 2 shows three sample equations from left to right in increasing complexity. The first equation has 10 characters and has no special symbols that do not appear on the keyboard. The second equation has 17 characters and no non-keyboard symbols. The third equation has 14 characters, two of which are special symbols. We expected that both components would have an effect on user performance. Increased length should increase time because additional characters in any modality would require more time to enter. Adding symbols that do not appear on the keyboard, such as ∑ and √, should only have a significant effect in the KB condition, because special symbols are no more difficult than normal symbols when speaking or writing. The length of each equation ranged from 10 to 18 characters.

Measures

The data from each session were collected at 30 frames per second by capturing the screen output and audio on a DV



$$f(x) = x^4 - 31 \quad \Big| \quad \frac{2x^2}{x+1} < |y| < \frac{x+5}{2} \quad \Big| \quad \int \left(\frac{u^7}{2} - \frac{\pi}{u^5}\right) du$$

Figure 2. Experimental stimuli as users saw them.

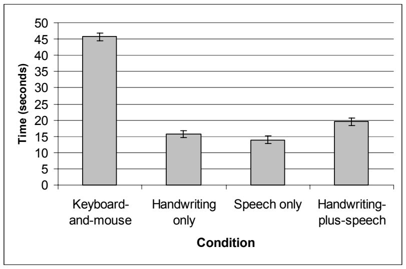


Figure 3. Average time in seconds per equation by condition. Error bars show 95% confidence interval (CI).

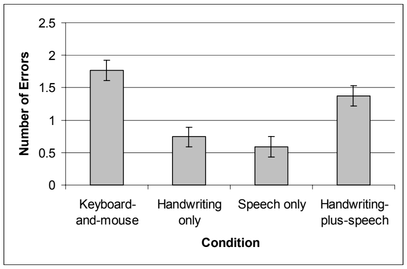


Figure 4. Mean number of errors made per equation by condition. Error bars show 95% CI.

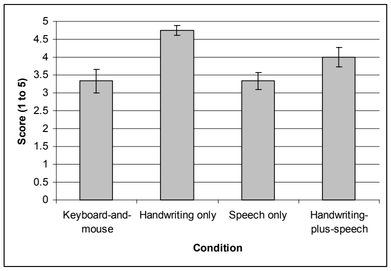


Figure 5. Post-test questionnaire rankings of each condition on a 5-point Likert scale. Error bars show 95% CI.

recorder. We later analyzed the videotape to extract the number of errors each participant made while entering each equation in each condition. An "error" was defined as when either the user submitted a completed equation with an incorrect character or the user acknowledged having made an error by correcting something previously entered. We also measured time for each participant to enter each equation in each condition. Finally, we have the users' preference ratings from both before and after the session.

RESULTS AND DISCUSSION

Qualitative Examples

Table 1 shows examples of an equation from each condition and a particular user's response to that equation. In the keyboard condition, the equation shown took the longest to complete out of all equations at 209 seconds for one user (note that it contains an error).

Speed

We ran a univariate ANOVA on time per equation considering the following factors: (1) participant as a random factor to account for the correlations between datapoints, (2) input condition and appearance of non-keyboard characters in each equation as fixed factors, and (3) the number of characters in each equation as a continuous covariate. This analysis yielded a significant interaction between condition and appearance of non-keyboard characters ($F_{3,1241}$=13.53, p<0.0005), and a significant main effect of the number of characters in the equation ($F_{1,1241}$=39.35, p<0.0005). Longer equations took more time to enter. The KB condition experienced a much larger slowdown due to appearance of non-keyboard characters than the other three conditions, which follows intuitively from the nature of the factor itself. Writing or saying a character that does not appear on the keyboard but is a common or natural mathematical notation is no more difficult than writing or saying actual keyboard characters. A planned contrast comparing the KB condition to the other three conditions showed a significant difference (t(1241)=34.91, p<0.0005), KB being slowest. Figure 3 shows the mean time in seconds for each condition.

Condition	Typeset Version	User Input
Keyboard-and-mouse	$\frac{1}{\lvert x\rvert+1}-\frac{x^2}{2}\le y$	$\frac{1}{/1/+1}-\frac{x^2}{2}\le y$
Handwriting only	$f(x)=5(y_1-y_2)$	$f(x)=5(y_2-y_1)$
Speech only	$\frac{y-4}{y^2-5y+4}=9$	"y minus four over y squared minus five y plus four equals nine"
Handwriting-plus-speaking (in parallel)	$\sum\left[c_k^2-2c_k-10\right]$	$\Sigma(c_k^2-2c_k-10)$ "sum of c subscript k squared minus two c subscript k minus 10 close parentheses"

Table 1. Samples of user input in the four conditions.

User Errors

We also performed a univariate ANOVA on errors per equation with the same factors as above. We found a significant three-way interaction between condition, length of the equation, and appearance of non-keyboard characters ($F_{4,1222}$=2.39, p<0.05), which implies that the length of the equation alters the relationship between condition and appearance of non-keyboard characters. However, because there were no significant two-order interactions, this relationship merits further exploration; here we will focus on the main effects of each variable. We found a significant effect of all three main factors: condition ($F_{3,1232}$=35.33, p<0.0005), the appearance of non-keyboard characters ($F_{1,1232}$=13.61, p<0.0005), and the length of the equation ($F_{1,1232}$=4.95, p<0.05). Longer equations and appearance of non-keyboard characters tend to result in more errors.

A planned contrast comparing KB to the other three conditions yielded a significant difference (t(1232)=8.17, p<0.0005), KB having more errors per equation. Figure 4 shows the expected mean number of errors per equation for each condition. The pattern in this graph matches the pattern of speed by condition shown in Figure 3. The relationship between condition and errors helps to explain why some conditions are slower than others. It is likely that the fact that the KB condition had the most errors per equation is an artifact of the input modality itself. Because the keys are close together on a keyboard, people often make typographical errors and must use the backspace key to correct them. The HWSP condition has about as many errors as the SP condition plus the HW condition combined because users did both modalities in HWSP.

Errors occurred on 52% of all equations. The SP condition was the least error-prone with errors on only 34% of equations performed in this condition, and a maximum of 5 errors on any one equation (compared to 58% in the KB condition, with a maximum of 15 errors). However, participants rated speech lowest in spite of its higher accuracy. They said later in informal interviews that they did not like the lack of feedback in the SP condition—with no visual reminder of what they had said, they did not feel confident in that modality.

User Preferences

We also asked users both before and after the session to rate the "suitability" and "naturalness" of each modality in the session for entering mathematics on computers. Users rated each modality's suitability on a 5-point Likert scale. On the pre-test questionnaire, there was no significant difference between the KB condition and the HW condition ($t(47)=-1.61$, $p>0.05$). However, on the post-test questionnaire, users rated the HW condition higher than the KB condition ($t(47)=4.49$, $p<0.0005$). Figure 5 shows the overall mean post-test questionnaire ratings of the four conditions.

Users were not restricted on how to perform the HWSP condition, meaning they could either use both modalities in parallel or perform one and then the other. Most users (77%) chose to use the two modalities in parallel for each equation. Of those who did *not* use the modes in parallel, they were split evenly between starting with handwriting or starting with speech. The way users performed the HWSP condition did not significantly affect their rating of that condition post-session ($F_{2,45}=0.34$, $p>0.05$).

FUTURE WORK

Having found evidence that handwritten equation entry is both faster and more preferred by users, we plan next to explore ways in which users naturally input the equations during the study. We plan to study our corpus of recorded data in order to yield design recommendations for future interfaces for mathematics input and to test alternative automated recognition methods.

Because users were not specifically instructed to avoid ambiguous equations, sometimes their input could be interpreted in several ways. For instance, saying "x plus 3 over 5" could mean "$x + (3 / 5)$" or "$(x + 3) / 5$". This was not addressed in this paper, but we plan to do so in future analyses of our corpus. Because users rated the handwriting-plus-speaking condition highly, we intend to explore the use of both streams of input (handwriting and speech) to help disambiguate recognition or input errors from separate recognition engines [6].

We will also look at errors per equation at a finer grain of detail and analyze the amount of time accounted for in each condition specifically by making and correcting errors. This could help normalize the error measure across conditions and allow subtle effects of complexity to emerge. Due to space limitations, we were unable to include an analysis of efficiency in this paper but we plan to pursue this further.

CONCLUSIONS

We have performed an empirical study on inputting mathematical equations using different modalities. The results from this study have indicated that the keyboard-and-mouse condition was significantly slower and more error-prone than the other three conditions. In addition, the more complex an equation (*e.g.*, longer and including special symbols) is, the more the KB condition slowed down, while the other three conditions did not experience as sharp a decline in speed with increased equation length. Finally, users in the post-test questionnaire preferred the HW condition more than the KB condition. The results from this study can help inform future design of more natural and efficient interfaces for mathematics.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation and the Pittsburgh Science of Learning Center. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NSF or PSLC. The authors would like to thank Chris Atkeson, Jiazhi Ou, Jacob O. Wobbrock, Adam M. Fass, Darren Gergle, Laura Dabbish, Ryan S. Baker, Angela Wagner, Anupriya Ankolekar, Scott Davidoff, Elsa Golden, Amy Hurst, and Ido Roll for their invaluable equipment, time, and support.

REFERENCES

1. Anderson, J.R., Corbett, A.T., Koedinger, K.R., and Pelletier, R.: Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences* 4 (1995) 167–207.
2. Blostein, D. and Grbavec, A. Recognition of Mathematical Notation. In *Handbook on Optical Character Recognition and Document Analysis*, Wang, P.S.P. and Bunke, H. (eds) (1996) 557–582.
3. Brown, C.M.L.: Comparison of Typing and Handwriting in "Two-Finger Typists." *Proceedings of the Human Factors Society* (1988) 381–385.
4. Locke, J.L. & Fehr, F.S. Subvocalization of Heard or Seen Words Prior to Spoken or Written Recall. *American Journal of Psychology* 85 (1972) 63–68.
5. Microsoft. *Microsoft Word User's Guide Version 6.0* (1993), Microsoft Press.
6. Oviatt, S. Mutual Disambiguation of Recognition Errors in a Multimodal Architecture. *Proceedings of the CHI Conference* (1999) 576–583.
7. Smithies, S., Novins, K., and Arvo, J.: Equation Entry and Editing via Handwriting and Gesture Recognition. *Behaviour and Information Technology* 20 (2001) 53–67.
8. Sweller, J.: Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science* 12 (1988) 257–285.