EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (2009)
C. Grimm and J. J. LaViola Jr. (Editors)

A Usability Evaluation of AlgoSketch: A Pen-Based Application for Mathematics

Theresa O'Connell[1 †] , Chuanjun Li[2], Timothy S. Miller[2], Robert C. Zeleznik[2], and Joseph J. LaViola Jr.[3]

[1] NIST, Gaithersburg, MD USA
[2]Brown University, Department of Computer Science, Providence, RI USA
[3]University of Central Florida, School of EECS, Orlando, FL USA

Abstract

AlgoSketch is a pen-based system for entering and editing mathematics. It supports interactive computation and early-stage mathematical algorithm design. In its paper-like environment, mathematical expressions can be entered anywhere on the page. Recognition and computational feedback are given in real time. We present results of a formative user evaluation of AlgoSketch, examining its applicability as a new interaction paradigm and users' overall experience with its mathematical entry, feedback, and computational support. Using a goal, question metric (GQM) framework, we evaluated AlgoSketch using efficiency, effectiveness, and satisfaction metrics. Logging data was supported by a post-questionnaire and anecdotal data. Results indicate acceptance of the AlgoSketch paradigm; strong potential for workplace utility, and a need for better mathematical expression recognition.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Evaluation/methodology

1. Introduction

Despite the computational power of computer hardware and mathematics applications, professional mathematicians and students often prefer pencil and paper. We posit a reason: pencil and paper afford fluid, direct specification of two-dimensional mathematical notations and supporting diagrammatic structures in arbitrary juxtapositions.

This work presents a formative user evaluation of AlgoSketch using the question metric (GQM) framework [BJ06] in terms of efficiency, effectiveness, and satisfaction, the dimensions of usability defined by the International Standards Organization [IOS]. AlgoSketch explores pencil and paper interaction, enhanced to provide integrated computational support for mathematics including multiple expressions, matrices, and algorithms [LMZL08, ZMLL08]. As general mathematical expression recognition is inherently ambiguous, AlgoSketch also implements an interactive recognition and feedback system to ensure a fluid workflow. We believe that this interactive feedback advances domain-specific mathematical sketching systems for a projected user base of mathematicians, analysts, teachers and students. Our long term goal is efficient, effective and satisfactory entering, editing and manipulating of mathematics.

2. Related Work

There have been several pen-based systems for mathematics. Chan and Yeung developed a simple pen-based calculator PenCalc [CY01] and Thimbleby and Thimbleby also developed a calculator with a gesture-based interface supporting animation for simple math calculations [TT05]. MathBrush [LMM*06, LLM*08] recognizes handwritten mathematics in order to drive a symbolic algebra system. MathJournal™ by xThink, Inc. can solve equations, perform symbolic computation, make graphs, and automatically segment multiple expressions, while Microsoft Math™ supports both numeric and symbolic computation of a single expression. LaViola's MathPad^2 system [LZ04, LaV07a, LaV07b] used a pen-based interface to let users

† Certain commercial equipment, instruments, materials, services or companies are identified in this paper in order to specify the experimental procedure. This in no way implies endorsement or recommendation by NIST.

create dynamic illustrations for exploring mathematical and physical concepts by combining handwritten mathematics and free-form drawings. The main difference between these systems and AlgoSketch is its real-time, write-anywhere, fluid interface for mathematical computation and, to the best of our knowledge, this paper presents the first formative evaluation of this type of interface.

3. The AlgoSketch System

AlgoSketch [LMZL08, ZMLL08] is a system for fluid pen-based entry and editing of mathematics with support for interactive computation. In its paper-like environment, multiple mathematical expressions and algorithms can be entered anywhere on the page. Mathematical expressions can be modified using simple deletion and dragging gestures with real-time recognition and computation feedback. It supports extended notations and gestures for controlling computational assistance, simplifying input, and entering algorithms.

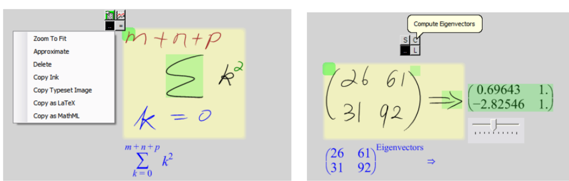

Figure 1: *Two screen shots of the AlgoSketch interface showing typeset interpretation of handwritten ink and examples of how green hover buttons can be used to invoke various mathematical operations.*

Our system currently supports entry of a solid range of basic math: a number of basic math symbols (including Greek letters and others such as ∞, $\mathbb{R}$, etc.), basic math relations ($<$, $\leq$, $\neq$, $\approx$, $\supset$, $\perp$, etc.), basic math operators ($+$, $-$, $\cdot$, $/$, $\wedge$, etc.), fractions written vertically ($\frac{a}{b}$), integrals, summations, roots, trigonometric and similar (log, ln, etc.) functions, and matrices. A typeset version of interpreted expressions, presented below the entered ink (see Figure 1), is updated continuously during sketching to facilitate identification of recognition errors [ZML07].

AlgoSketch supports additional input extensions for graphing, simplifying, and numerically approximating entered expressions. A looping gesture [ZMLL08] drawn from a recognized expression creates a graph of it. Extended notations, $\rightarrow$ and $\Rightarrow$, used at the end of expressions, request simplification and numeric approximation, respectively. In an evaluated expression, variables which are defined elsewhere on the page will be substituted. AlgoSketch supports a range of editing interactions: scribbling over ink to delete it, circling ink to select it for dragging or copying and undo, redo, and a menu to choose alternate interpretations for entered characters. Widgets are also attached to each written expression to provide additional functionality. Green squares are drawn under the ink on the display; hovering the pen tip over the square without touching the screen causes a cluster of buttons to appear to the upper-right (see Figure 1). If the pen moves directly towards the buttons, they can be pressed and interacted with to display menus, perform functions, etc. This design means that every pixel can be the start of an ink stroke.

4. AlgoSketch Evaluation

The purpose of the AlgoSketch formative usability evaluation was to gain early insight into (1) the applicability of the paradigm of entering and computing math on a tablet computer with a pen and (2) subjects' experience with entry and computation tools for matrices. Our ultimate goal is user efficiency, effectiveness and satisfaction when employing AlgoSketch to enter, edit and manipulate mathematics.

4.1. Subjects

There were six subjects, four male and two female. They were designated U1A, U2M, U3S, U4M, U5S, and U6S where A denoted a mathematician analyst; M, a professional mathematician; and S, an undergraduate mathematics student. U1A was a cryptology analyst; his workplace responsibilities included linear algebra. U2M did mathematics research and taught mathematics at the university level. U4M, an engineer in informatics and mathematics, was developing a mathematical calculating tool. Students' ages ranged from 19 to 21; professionals from 24 to 56. All subjects had normal vision; none had color deficiencies in vision. All were right-handed. None had experienced AlgoSketch before the study. All were familiar with electronic math tools such as Mathematica. All had medium to expert self-rated computer expertise. None reported expertise in tablet computer use.

4.2. Apparatus

Each subject interacted with AlgoSketch using an Compaq tc4400 Tablet PC with 1.8Ghz Intel Centrino Duo and 512 MB RAM. Resolution of the display was set to 1024x768. Subjects also used a desktop workstation for online surveys. Both the tablet PC and desktop workstations were connected to a local area network at a speed of 100 Mbs. A connection between the two PCs was established via RealVNC. This connection fed a display of the tablet PC's screen to the desktop workstations' second monitor. The second monitor was not visible to subjects; it was angled away from subjects, toward the observers. In this way, usability engineers (UE) observed subjects' interactions with AlgoSketch from an unobtrusive distance, while maintaining a full view of the subjects.

In order for all subjects to experience the same experimental environment, subjects were required to work with the

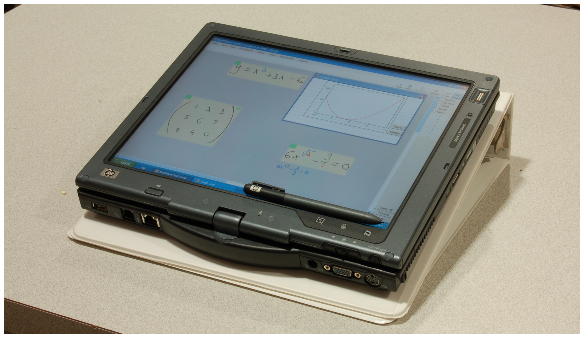

Figure 2: *A Compaq tc4400 Tablet PC, running AlgoSketch, set at a 15 degree angle on a 2 inch high binder.*

tablet PC in the same configuration. Investigation with surrogate users resulted in using a two-inch binder as a consistent slanted stand for the Tablet PC (see Figure 2). Because all subjects were right-handed, there was no need to rotate the tablet to accommodate left-handedness.

Overhead lighting in usability laboratories usually simulates an office environment. This lighting caused glare on the tablet's screen. To reduce glare, but maintain a uniform lighting level, we dimmed the light to a consistent average range of 440 - 490 lux (44 - 46 foot-candles) across all subjects except U2M. Lighting was dimmed to a level comfortable for U2M, roughly one quarter of the brightness/intensity of standard lighting conditions (a range of 105-148 lux or 9.7 - 14 foot-candles). Otherwise, the evaluation environment was consistent across all subjects.

4.3. Experimental Tasks

Subjects played the role of a mathematician who had to enter mathematical expressions into a computer. They were told they might have to refine entries. Subjects were not constrained to using a particular technique when AlgoSketch offered more than one way to accomplish a goal. This freedom to use their own preferred work styles accommodated human variation in input. It supported study of subjects' perceptions of recognition accuracy. It shed light on subjects' preferences among AlgoSketch options and made it possible to study the degree to which AlgoSketch supported the variety of work styles expected within AlgoSketch's varied intended user population.

Six algebraic problems (P1-P6) simulated subjects' workplace and academic tasks. P1-P4 required subjects to enter math expressions and compute. P2 required graphing. P3 and P4 required creating and computing matrices. P5 required subjects to find values for the parameters of an ellipse contained inside a rectangle. P6 required subjects verify a complex exponential identity. Subjects had to verify that recognition was correct. They had to work with a range of key functionality to enter and complete any of the problems. Thus subjects experienced every key functionality, enabling them to give informed ratings on each.

4.4. Usability Metrics

UEs who had not been involved in the design of AlgoSketch identified metrics using a hierarchical GQM framework that proceeds from goal setting to question definition to metrics identification [BCR94]. From a GQM perspective, the role of metrics is to track conformance to goals. This results in purposeful and relevant metrics [BJ06]. GQM goal statements formally set out a goal's purpose, motivating issue, object and viewpoint [BCR94]. The viewpoint in this study was the user's. We addressed five goals that we categorized according to efficiency, effectiveness and satisfaction. This approach put the burden for usability on AlgoSketch and gave subjects the roles of evaluators. Below, we set out our goals, questions and metrics, stating the goals in GQM format.

Efficiency goal: Using AlgoSketch, enable specification of mathematical expressions by subjects. This goal led to two questions: **To what degree can subjects interactively specify a math expression before making any modifications while solving problems?** For each subject, we counted input characters, symbols, and strokes required to input polynomials, fractions, equalities, inequalities, roots, summations, and integrals. We counted instances of gesture and button usage. **To what degree can subjects interactively edit a math expression?** To study efficiency during interactive editing of ink strokes, we counted appends without crossing out and after crossing out; deletions without appending; and circle-and-move gestures. We counted choices from the alternates list. We counted events when subjects undid and redid input as well as extensions to horizontal lines in square root symbols.

Efficiency goal: Using AlgoSketch, enable computation of mathematical expressions by subjects. Two questions resulted. **To what degree can subjects perform calculations on the expressions entered?** We counted $\rightarrow$ and $\Rightarrow$ notations. **To what degree can the subject graph?** We counted uses of the graphing functionality.

Efficiency goal: Using AlgoSketch, enable specification of matrices by subjects. This goal led to one question. **To what degree can subjects specify advanced matrix operations?** We counted matrices entered; components in each matrix entered; and matrix green button selections.

Effectiveness goal: Using AlgoSketch, enable recognition of careful input by subjects. This goal led to one question. **If a subject makes an effort to input carefully, to what degree can AlgoSketch recognize that input?** We counted misrecognition correction attempts.

Satisfaction goal: Provide a satisfactory experience with AlgoSketch from the subjects' point of view. This goal led to three questions. **To what degree were subjects satisfied with the paradigm of using pen-based computing to do math?** A multiple-choice question disclosed subjects' input preferences, after using AlgoSketch. We also asked subjects to compare AlgoSketch to their current tools. **To what degree were subjects satisfied with the experience of using targeted AlgoSketch functionality?** For each type of logged interaction, the survey collected satisfaction ratings over dimensions such as usefulness, meeting expectations and ease of use. **To what degree are subjects willing to adopt AlgoSketch for their work?** Five questions on insertion readiness addressed this issue.

Objective and subjective data were collected on subjects' interactions with key AlgoSketch functionality. Because each GQM goal was user-centered, the metrics derived were user-centered. The evaluation applied 71 metrics, many derived from metrics for visualizations [OC08], but customized to the unique functions of AlgoSketch. We used programmatic logging to count interactions with key functionality. A post-survey quantified satisfaction data in 1-7, worst-to-best, Likert-like scales. This survey contained 36 questions designated Q1-Q36. Thirty of these questions were scalar. Five were open-ended to elicit anecdotal data in which subjects explained their ratings. One was multiple choice.

4.5. Experimental Procedure

This evaluation required a methodology that applied usability best practices [OC08], but customized them to be sensitive to the unique characteristics of human interaction with a pen-based tablet computer. The study was user-centered, focusing on the user experience. There were six sessions, each with one subject. Activities occurred in the same sequence across all sessions.

Sessions started with the UEs explaining that the goal of a prototype is to place a subset of a tool's functionality before subjects for evaluation. This instruction was necessary to ensure that subjects did not transfer negative impressions of a prototype's limitations to their evaluation of AlgoSketch. An online demographic survey verified that subjects fit the intended user profile; an ancillary goal was to document experiences that related to using a tablet computer for math.

A 50-minute self-paced and self-administered training session familiarized subjects with the tablet. In videos, it introduced AlgoSketch's key concepts and functionality:

- Write a complex math expression
- Delete a character or set of characters using a scribble
- Correct character recognition using the alternates menu
- Correct part of an expression by scribbling it out and rewriting it
- Move characters around using a circle and move gesture
- Invoke menu items from green buttons
- Compute symbolic and numerical results using $\longrightarrow$ and $\Rightarrow$ notations
- Create a graph
- Enter a matrix
- Perform matrix calculations using the green button
- Insert a space in a matrix using gestures.

Subjects had 50 minutes to work on the set of six problems. The goal was that subjects work at their own pace, yet experience AlgoSketch's key functionality. All subjects received the problems in the same order. UEs handed individual problems to subjects one at a time, on separate sheets of paper, as subjects either completed problems or elected to move on without completing them. Subjects who did not receive all of the problems were not aware of this. After the problems, all subjects except U4M were encouraged to explore AlgoSketch, using it as they might in their own workplaces. U4M had explored extensively during problem-solving. Exploration enabled subjects to make informed judgments on insertion readiness. UEs took time-stamped notes on subjects' experiences with AlgoSketch throughout the sessions, recording factors such as their comfort level. Logging ran throughout all interactions with AlgoSketch.

A post-evaluation survey collected and quantified subjective feedback about experiences with AlgoSketch. Four subjects chose to "think aloud" about motivations for their ratings as they completed the survey. Afterwards, UEs interviewed each subject individually. Subjects' comments were later incorporated into the analysis of survey responses.

Subjects then viewed a two-minute video showing functionality for sketching of mathematical algorithms that was not yet robust enough for user evaluation. They completed a five-question 1-7, worst to best, Likert-like scale survey on their expectations for impacts of this functionality on their own work.

5. Results

Logging verified that subjects had experienced the functionality that they evaluated. Anecdotal data, subjects' statements and UEs' observations, provided support for the quantitative data. The analysis of this data along with usability principles informed recommendations. Subjects' recommendations were used only when they did not violate usability principles. In this section, we present examples of strengths and needs for improvement uncovered in the evaluation.

5.1. Acceptance of the AlgoSketch Paradigm

Subjects' ratings and comments reflected acceptance of the AlgoSketch paradigm. Pen-based input to AlgoSketch aligns with their experience of writing mathematical expressions on paper. The application of established behaviors and expectations for writing on paper to transferred to AlgoSketch.

This alignment emerged as the principal strength of AlgoSketch.

Behavior Transfer. In one sense, AlgoSketch's main competition is a piece of paper. In transferring established behaviors of working on paper to working with AlgoSketch, subjects were dissatisfied when the transition was not seamless. Conversely, behaviors that transferred easily promoted learnability and satisfaction. For example, some AlgoSketch command gestures, such as scribbling out, mirrored gestures typically made on paper.

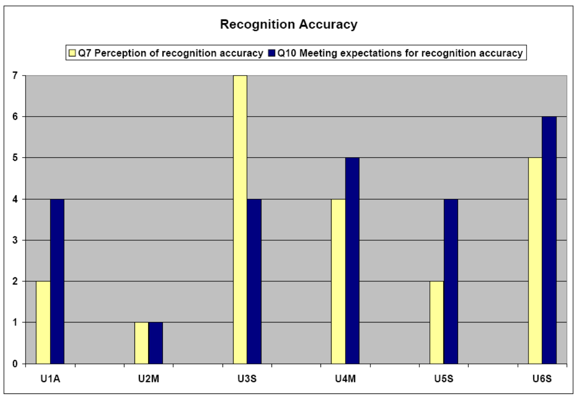

Figure 3: *Younger subjects gave higher AlgoSketch acceptance ratings than older subjects.*

Behaviors that subjects tried unsuccessfully to transfer from paper to the tablet included writing quickly at the expense of quality. Subjects consistently spurned the undo button to exhibit a behavior more natural for paper than a computer screen, i.e., deleting then entering. Building on their paper-based experience, subjects maintained a record of their prior work, which caused system response time to slow.

Expectations Transfer. Meeting users' expectations increases usability. In transferring paper-based behaviors to AlgoSketch, subjects also transferred expectations that conflicted with AlgoSketch functionality, resulting in them adopting new behaviors, which caused dissatisfaction. First, they had a priori assumptions about when their handwriting would and would not be recognized, which tended to be more optimistic then what AlgoSketch supported, requiring them to write more carefully. Second, users expected the interpretation of existing non-local characters and expressions not to change as new ink is added. This expectation was violated, for example, when an open parenthesis, initially interpreted as a '1', switched to '(' when its matching close parenthesis was entered. Two users, U3S and U4M, recognized the benefit of waiting until the whole expression was entered before making corrections, although the inclination of all but one of the users was to correct immediately, counter to the tutorial's recommendation. Last, subjects expected no adverse effects from stray marks. Stray marks resulted from tapping on small widgets such as an outer sector of a green button menu. During dragging and scrolling, the pen sometimes slipped, making a stray mark. Often, subjects were unaware of stray marks. Other times, because of their paper-based experiences, they paid them no attention. Subjects were not aware that the marks interfered with computation, puzzling subjects who could not compute correctly in their presence. It is interesting that U3S who realized the effect of stray marks and took control by erasing them, generally gave high satisfaction ratings, averaging 5 over post-evaluation ratings. User control promotes usability. As an example of a way to support users more generally, the recognizer could be tuned to ignore small marks made with low pressure, since this would be consistent with paper and pencil behavior of reinforcing faint marks by overdrawing.

Age Differences. Interestingly, there was evidence of age differences in acceptance of the AlgoSketch paradigm, even in this small sample. The four younger subjects' ratings were generally higher and their comments more positive than those of the two older subjects, U1A and U2M (see Figure 3). A belief that the AlgoSketch prototype will develop into a desirable tool was particularly evident in younger subjects' comments, with their satisfaction comments appearing to pertain to a future, mature version of AlgoSketch. Responses to Q17, preferred choice for entering mathematical expressions, showed that U1A and U2M preferred keyboard input, U3S and U6S preferred paper, and U4M and U5S preferred pen entry. It is not clear whether these differences should be attributed to age, to generation, or to experience.

5.2. Specifying Mathematical Expressions

Logging disclosed that each subject successfully specified mathematical expressions. Each entered polynomials, fractions, equalities, inequalities, roots, summations and integrals at least once, usually multiple times.

The tablet's size provides portability, but there is a tradeoff in its small screen. AlgoSketch attempts to maximize input area through a minimalist design strategy. For example, the green button effectively occupies no space since its menus appear only when needed. This efficient canvas management reduces effort by enabling access to needed functionality without leaving the area where attention is currently focused. However, the program additionally has a conventional menu and toolbar to provide global commands and aid gesture and functionality discoverability, and the study uncovered needs to be able to hide, show, float, and dock them to increase the workspace. When there wasn't sufficient canvas to spread out work, subjects experienced unintentional merging of math grouping areas or matrices. U6S requested "some way to specify that the expression should be treated independently and not be sucked in to other expressions." Locking input will help users organize screen space and avoid undesired mergers. Subjects also requested the ability to scale down portions of their work to maximize workspace.

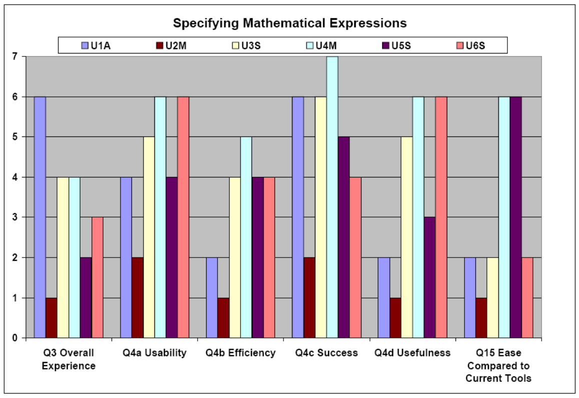

Figure 4: *Subjects' ratings on specifying mathematical expressions varied.*

Figure 4 shows subjects' ratings on specifying mathematical expressions. Most ratings for overall experience (Q3) were in the neutral to low range. U2M cited poor recognition as a reason behind her rating of 1. We believe recognition issues to be the largest factor behind the other low ratings for this question as well. Most ratings were neutral to negative again for the efficiency of specifying a mathematical expression in a reasonable amount of time (Q4b). U6S explained his neutral rating, "Simple expressions are fine, but when you get to more complex expressions..., that little mistake makes it interpret things differently. Usability problems make it like a task." This disproportionate usability issue might be attributed to several things, such as misrecognized characters causing cascading structural misrecognitions, more complicated math having more complicated and numerous interrelationships to scan for errors, the increased likelihood of a structural error, or the difficulty of spatial planning. However, four subjects responded positively to Q4c, on success in specifying a mathematical expression, and three responded positively and three negatively to Q4d, on the usefulness of this functionality. Q15 asked subjects to compare the ease of entering expressions using AlgoSketch to that of current workplace tools. U6S gave a very low rating of 2, but praised the speed of inputting math expressions, "You gain the ability to do things quicker."

Editing. Logging tracked editing and modifications. Most subjects employed all methods except redo which no subject intentionally exercised. Subjects were successful in executing all editing functionality, but needed alternate characters were not always available. Appending after crossing out was the most often used modification method. The next most often used methods across all subjects included appending without crossing out and deleting without inputting. Consistent with the phenomenon of subjects transferring paper-based behaviors to AlgoSketch, five subjects employed gestures more often than buttons to activate functions. Only U2M used buttons more often than gestures.

Logging, comments and observations disclosed that AlgoSketch's flexibility in editing options promotes effectiveness. AlgoSketch is also strong in giving subjects obvious and immediate feedback on input and a variety of ways to correct errors. It is a strength of AlgoSketch that it brings errors to users' attention. For example, subjects comfortably checked typeset recognition feedback for errors. By promoting subject efficiency, AlgoSketch error management strategies promote usability.

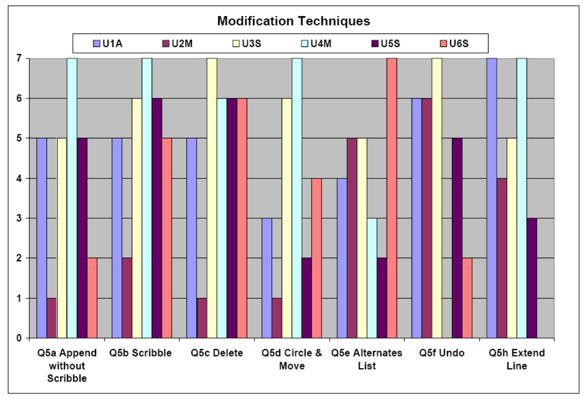

Figure 5: *Subjects rated eight modification techniques.*

Subjects rated their satisfaction with eight modification techniques (see Figure 5). No subjects experienced redo sufficiently to produce informed ratings. Thus, none are reported. Most subjects did not notice the delete button, motivating a recommendation to move it to a more obvious location in a redesigned toolbar. The location of the undo button was also problematic, leading to sparse use (33 uses across all subjects and all activities). Its location next to the delete button introduced the possibility of errors.

Overall ratings for error management were generally low. U6S said the reason was a need for too much user effort, "Simple expressions are fine, but [with] more complex expressions, simple mistakes take a long time to correct."

5.3. Computation on Mathematical Expressions

Logging disclosed that all subjects were able to perform computations using the $\rightarrow$ and $\Rightarrow$ symbols. There were 215 instances over the six sessions. Despite problems writing these symbols, subjects comfortably performed computations with them because using them met their expectations.

Logging showed that all subjects computed graphs successfully and developed preferences on how to initiate graphing. Three subjects preferred to graph by button and three by gesture, attesting to a usability strength of AlgoSketch in accommodating work style alternatives. U6S, who preferred gestures, found "The ability to graph by doing this motion is pretty natural. I won't forget how to do it." Q14 asked subjects to rate the ease of graphing. Four subjects gave the highest rating of 7. U2M rated ease at 4, U3S at 3. U6S said, "[graphing] couldn't be any more simple".

Most subjects generally gave high ratings on two aspects of AlgoSketch's computation capabilities. Q11 asked them to rate the degree to which AlgoSketch met their expectations when it showed them computation results. Q12 asked them to rate the ease of performing computations on mathematical expressions. With the exception of U2M, most ratings were high.

5.4. Specifying Matrices

Logging showed that all subjects successfully entered matrices. Over all session activities, subjects completed matrices 98 times. They entered a total of 444 matrix components and used the green hotspot matrix menus 44 times.

The principal problems subjects encountered when specifying matrices were entering parentheses and columns of numbers with the precision that AlgoSketch required. AlgoSketch required straight columns and curved parentheses that were long enough to completely encompass the columns. The human eye and brain do not need this precision, so it diverged from subjects' usual way of writing matrices on paper. U2M requested an onscreen grid. This strategy will also address parentheses length. Alternatively, it seems likely that the recognizer could be improved to not have this restriction. Accommodating square brackets will give users more control over input styles.

Q13 asked subjects to rate the ease of specifying a matrix. Problems entering matrices motivated one neutral score (U6S, 4) and three in the negative range (U2M, 1; U3S, 2; U4M, 3). Two were positive (U1A, 5 and U5S, 6). Despite their problems with matrices, there was no discomfort noted while subjects worked with matrices.

5.5. Recognition Quality Effects

Logging showed that subjects accessed the alternates list 270 times during evaluation problem solving and 402 times when training and exploration are added. During problem solving, U4M was the least frequent user of alternates at 21 accesses and U2M the most frequent at 155. It was not possible to ascertain from logging when careful input resulted in recognition or when sloppy input caused misrecognition, but observations and subjects' comments indicated careful, neat input was necessary.

Subjects displayed a wide variety of writing styles. They did not normally write mathematical expressions carefully on paper, and did not want to do so on the tablet. There were handwriting inconsistencies within individual subjects. Subjects at first attributed misrecognition to the poor quality of their handwriting, but later blamed the system for failure to recognize good input. Subjects expressed low expectations for recognition accuracy.

Logging did not identify which letters were misrecognized. However, logging did record which characters were selected from the alternates menu. These spanned a wide range encompassing basic math symbols (including Greek letters), parentheses, digits and Roman letters. It must be noted that this is a rough indicator of which input was not recognized because of at least three confounding factors. (1) The desired character was not always in the alternates menu. (2) Subjects did not always use the alternates menu to resolve misrecognition. (3) Some character selections reflect the content of the evaluation exercises rather than the wide variety of possible input. As recognition accuracy increases, usability will also increase. To expand the base of recognizable input, there is a need for the recognizer to support diverse ways of writing characters, collected from writing samples from the wide range of intended users.

Often, the alternates list did not contain the character a subject sought. This required them to re-enter characters, sometimes repeatedly, until they were recognized. Therefore, subjects wanted to be able to specify alternates that did not appear in the alternates list. U6S observed, "I know there are options about what character you're writing, but not what you intend it to be. That would be advantageous to have." The high alternates-list usage across all subjects led to a recommendation for an abridged character map, containing only the characters AlgoSketch supports, to give users an overview of all their character options when the desired option does not appear in the alternates list. Both of these features in fact were ones the AlgoSketch authors wanted to add but did not get time to do before the study.

Q7 asked subjects to rate AlgoSketch's recognition accuracy and Q10 asked them to rate how well typeset met their expectations for accuracy. Comments showed that perceived accuracy impacted satisfaction with AlgoSketch. U1A said "Not recognizing the characters stands in the way of it being useful." The two older subjects, U1A and U2M, citing poor recognition, gave negative to neutral ratings for AlgoSketch's insertion readiness.

5.6. Expectations Survey

After a short video on future algorithm sketching functionality, subjects gave agreement ratings to five statements in an expectations survey. On average, subjects rated their expectation that the functionality would allow them to accomplish tasks more quickly at 4.7, increase their productivity at 4.3, enhance their effectiveness at 4.6, facilitate their work at 4.3, and that they would find it useful in their work at 4.6.

6. Discussion

Over all evaluation activities, subjects input 9,867 strokes: U1A, 2,039; U2M 1,242; U3S 1,010; U4M, 1,639; U5S, 1,789; and U6S, 2,148. Logging and observations disclosed that subjects were able to enter and edit polynomials, fractions, equalities and inequalities, roots, summations, integrals, and matrices. They performed calculations on expressions they had entered. They easily used the matrix menu to perform various matrix operations. They easily graphed polynomials in one variable. Subjects were able to exercise all of AlgoSketch's key functionality.

Satisfaction varied across AlgoSketch's wide range of functionality. However, with the exception of U2M, overall satisfaction ratings were mostly in the range of 5-7. Subjects' experiences with input misrecognition and the small tablet screen size appeared to be the primary impediments to higher survey ratings. The interview gave subjects the opportunity to comment on any aspect of their experience with AlgoSketch that they wanted. With the exception of U2M, subjects' comments were positive, predicting workplace utility and acceptance of a mature AlgoSketch.

The evaluation resulted in 52 usability recommendations, typical of a prototype. These recommendations were categorized into essential changes that would have a major positive impact on user efficiency, effectiveness, and satisfaction (high priority), needed changes that would have a positive impact but are not critical (medium priority), and non-essential changes that would improve the overall user experience (low priority). Of the 35 high priority recommendations (see the complete list in the Appendix), many of them involved creating a toolbar with access to basic AlgoSketch functionality (e.g., cut, copy, paste, undo, redo). We believe a toolbar that provides information on how to perform these different AlgoSketch gestures (i.e., a GestureBar [BZW*09]) would go a long way to improving the UI. Other high priority recommendations included better overall recognition to accommodate different user writing styles and user control over inputting, accessing functionality and managing the ink canvas (e.g., scaling parts of the ink canvas to maximize the workspace). Ease of use and flexibility can also be increased by accommodating a wider variety of work styles and workflows. Such improvements will raise user efficiency, effectiveness and satisfaction.

7. Conclusion

This study showed that the AlgoSketch prototype had many usability strengths and that a mature AlgoSketch will have a strong potential for workplace utility. Subjects' comments indicate that, to succeed, AlgoSketch must offer pen entry that is as easy as writing on paper. U2M said, "If I could enter into a computer using traditional paper that would be the optimal." AlgoSketch offers a paradigm for entering mathematical expressions that builds on mathematicians' experiences using paper. The evaluation showed that paradigm is natural, powerful and promising. In the words of U4M, "It is a very good tool… It can be a very useful tool."

Acknowledgements

This work was supported in part by Brown University, the Intelligence Advanced Research Projects Activity (IARPA), the National Institute of Standards and Technology (NIST) and the University of Central Florida.

References

[BCR94] BASILI V. R., CALDIERA G., ROMBACH H. D.: The goal question metric approach. In *Encyclopedia of Software Engineering*. Wiley, 1994.

[BJ06] BERANDER P., JÖNSSON P.: A goal question metric based approach for efficient measurement framework definition. In *ISESE '06: Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering* (New York, NY, USA, 2006), ACM, pp. 316–325.

[BZW*09] BRAGDON A., ZELEZNIK R., WILLIAMSON B., MILLER T., LAVIOLA JR. J. J.: Gesturebar: improving the approachability of gesture-based interfaces. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems* (New York, NY, USA, 2009), ACM, pp. 2269–2278.

[CY01] CHAN K.-F., YEUNG D.-Y.: Pencalc: A novel application of on-line mathematical expression recognition technology. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition* (September 2001), pp. 774–778.

[IOS] International organization for standardization, ergonomic requirements for office work with visual display terminals (vdts) - part 11: Guidance on usability. 1998.

[LaV07a] LAVIOLA J.: Advances in mathematical sketching: Moving toward the paradigm's full potential. *IEEE Computer Graphics and Applications 27*, 1 (January/Febuary 2007), 38–48.

[LaV07b] LAVIOLA J.: An initial evaluation of mathpad^2: A tool for creating dynamic mathematical illustrations. *Comput. Graph. 31*, 4 (2007), 540–553.

[LLM*08] LABAHN G., LANK E., MARZOUK M., BUNT A., MACLEAN S., TAUSKY D.: Mathbrush: A case study for interactive pen-based mathematics. In *Proceedings of the 5th EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (SBIM 2008)* (June 2008), pp. 143–150.

[LMM*06] LABAHN G., MACLEAN S., MIRETTE M., RUTHERFORD I., TAUSKY D.: Mathbrush: An experimental pen-based math system. In *Challenges in Symbolic Computation Software* (2006), Decker W., Dewar M., Kaltofen E., Watt S., (Eds.), no. 06271 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

[LMZL08] LI C., MILLER T., ZELEZNIK R., LAVIOLA J.: Algosketch: Algorithm sketching and interactive computation. In *Proc. of the 5th EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (SBIM 2008), pp. 175-182* (2008).

[LZ04] LAVIOLA J., ZELEZNIK R.: Mathpad^2: A system for the creation and exploration of mathematical sketches. *ACM Transactions on Graphics 23*, 3 (Aug. 2004), 432–440. (Proceedings of SIGGRAPH 2004).

[OC08] O'CONNELL T., CHOONG Y.: Metrics for measuring human interaction with interactive visualizations for information analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), pp. 1493–1496.

[TT05] THIMBLEBY W., THIMBLEBY H.: A novel gesture-based calculator and its design principles. In *Proceedings 19th BCS HCI Conference* (2005), MacKinnon L., Bertelsen O., Bryan-Kinns N., (Eds.), vol. 2, British Computer Society, pp. 27–32.

[ZML07] ZELEZNIK R., MILLER T., LI C.: Designing UI techniques for handwritten mathematics. In *Proceedings of the 4th EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (SBIM 2007)* (Aug. 2007).

[ZMLL08] ZELEZNIK R., MILLER T., LI C., LAVIOLA J.: Mathematical sketching with fluid support for online computation. In *Smart Graphics 9th International Symposium, SG 2008* (Rennes, France, August 2008), pp. 20–32.

Appendix

The 35 high-priority recommendations yielded by the study:

- Train the system to accept a wider variety of handwriting input styles.
- If there are hints for character formation that will raise recognition rates, give these hints in the AlgoSketch tutorial.
- When introducing new gestures to AlgoSketch, use gestures that are unlike those users employ when writing.
- If possible, provide access to a list of expressions that have been recognized.
- Create a toolbar that has a default location in the area below the alternates menu. If such a toolbar is added, give users the ability to locate it in a position of their choosing.
- In the toolbar, provide dividers between categories of functionality. Provide a double divider after the clear all button.
- Give users the ability to display the toolbar as shown in Figure 10 or to configure it to display as a horizontal bar.
- Create cut, copy and paste buttons for the toolbar.
- Create zoom in and zoom out buttons for the toolbar.
- Move the undo button to the toolbar. Position the undo button to the left of a redo button.
- Add a redo button to the toolbar. Group this button with the undo button, locating it to the right of the undo button.
- Move the delete button to toolbar.
- Move the clear all button to the toolbar.
- Give users the ability to scale user-designated parts of their work, making their work appear smaller or larger.
- Give users the ability to easily designate the area that they will zoom on, perhaps by drawing a circle around it.
- In the matrix work area, change the name of the More Options menu in the right green hotspot to More Matrix Operations. Work with users to identify an appropriate name for the left hotspot More Options menu that appears in both the work area and the math work area.
- Give users the ability to close the green hotspot menus.
- Give users the ability to lock individual yellow math grouping areas.
- Give users the ability to lock individual matrices.
- Do not make the recognition heavily dependent on the direction in which characters are written.
- Provide an abridged character map that contains only all the characters supported by AlgoSketch. At the bottom of the alternates list, provide a link to the character map. Label this link More Alternates.
- In every error message, explain the error and provide a remedy in words that non-technical users can understand.
- If AlgoSketch has the ability to identify an undefined variable, when there is an undefined variable, provide a pop-up message, Variable X is undefined. Do you want to continue evaluating this expression? Provide Yes, No buttons.
- In addition to filling in the background of the icon for the active input mode with blue, outline this icon with a heavy black line. Consider giving this button a 3D look to stress the active mode.
- Change the default look of the cursor when the user is in drawing mode.
- Develop software that will interpret a gesture or a set of gestures to activate scroll-wheel functionality.
- In the training, instruct users that the system will attempt to recognize stray marks and that they should take care not to make stray marks on the screen.
- In the tutorial, explain the effect of drawing a long line next to the vertical scroll bar.
- Give users the ability to select, copy, and paste evaluation results as handwriting/ink which can be used in all of the same ways as other handwriting/ink.
- Give users the ability to designate a screen location where a paste will appear. AND/OR When a user executes a paste, surround the paste with a circle so that the user can drag the paste to the desired location.
- Replace the letters, S, C and L, with full menu names.
- For specifying matrices, recognize both square brackets and parentheses.
- Give users the option of activating a grid over the AlgoSketch canvas.
- In the tutorial, stress that users must enter matrices neatly.
- In the matrix menu, replace the words, No Matrix Operation with the words, Cancel Matrix Operation.