EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2006)
Thomas Stahovich and Mario Costa Sousa (Editors)

An Initial Evaluation of a Pen-Based Tool for Creating Dynamic Mathematical Illustrations

Joseph J. LaViola Jr.

Brown University, Department of Computer Science, USA
Email: jjl@cs.brown.edu

Abstract

MathPad^2 is a pen-based application prototype for creating mathematical sketches. Using a modeless gestural interface, it lets users make dynamic illustrations by associating handwritten mathematics with free-form drawings and provides a set of tools for graphing and evaluating mathematical expressions and solving equations. In this paper, we present the results of an initial evaluation of the MathPad^2 prototype, examining the user interface's intuitiveness and the application's perceived usefulness. Our evaluations are based on both performance and questionnaire results including first attempt gesture performance, interface recall tests, and surveys of user interface satisfaction and perceived usefulness. The results of our evaluation suggest that, although some test subjects had difficulty with our mathematical expression recognizer, they found the interface, in general, intuitive and easy to remember. More importantly, these results suggest the prototype has the potential to assist beginning physics and mathematics students in problem solving and understanding scientific concepts.

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

1. Introduction

MathPad^2 (see Figure 1) is a pen-based, Tablet PC application prototype for creating dynamic illustrations used for exploring mathematics and physics concepts [LZ04]. The fundamental technology behind MathPad^2 is mathematical sketching, a pen-based gestural interaction paradigm for mathematics problem solving that derives from the familiar pencil-and-paper process of drawing supporting diagrams to facilitate the formulation of mathematical expressions; however, with mathematical sketching, users can also leverage their physical intuition by watching their hand-drawn diagrams animate in response to continuous or discrete parameter changes in their written formulas [LaV05]. Diagram animation is driven by associations that are inferred, either automatically or with gestural guidance, from handwritten mathematical expressions, diagram labels, and drawing elements.

The essential goal in developing the MathPad^2 user interface was that it be as similar and fluid as pencil and paper, since mathematics and physics problems are often solved using this medium. Thus, we did not want to use any additional hardware (e.g., a modifier key or stylus button) or software (e.g., buttons) modes. Instead, we wanted all interaction to be derived from using digital ink. We developed a gestural user interface for invoking different operations in MathPad^2 because we wanted users able to work as fluidly as possible with the mathematics and drawings they create. We wanted to explore whether our choice of gestures, which by themselves are not part of pencil-and-paper interaction, are thought of as intuitive or at least complimentary to pencil and paper.

Given the foundations for MathPad^2, we performed an initial usability evaluation to gauge users' performances and reactions to the prototype to validate its design and potential benefit and determine if further, more in-depth studies are needed. More specifically, we are interested in how easy it is for users to use MathPad^2 with only a visual demonstration of how to invoke gestural operations, and in how many mistakes they make in performing various MathPad^2 tasks. We are also interested in how well subjects remember various gestural commands, since this is a good indicator of intuitiveness. Using interface satisfaction [CDN88] and perceived usefulness [Dav89] questionnaires, we are addition-

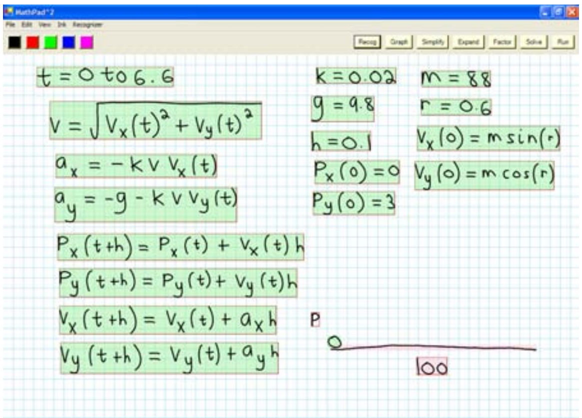

Figure 1: *A mathematical sketch, created in MathPad^2, illustrating how air drag affects a ball's 2D motion. Associations between mathematics and drawings are color-coded.*

ally interested in whether subjects would use mathematical sketching in their work and why.

2. Related Work

The idea of using computers to create dynamic illustrations of mathematical concepts has a long history. One of the earliest dynamic illustration environments was Borning's ThingLab, a simulation laboratory environment for constructing dynamic models of experiments in geometry and physics, that relied heavily on constraint solvers and inheritance classes [Bor79]. Other systems such as Interactive Physics™ and The Geometer's SketchPad™ also let the user create dynamic illustrations; these systems are all WIMP-based (Windows, Icons, Menus, Pointers) resulting in a significant amount of mode switching and loss of fluidity within the interface. In addition, they do not allow the user to write handwritten mathematics to create these illustrations. Because MathPad^2 uses handwritten mathematical expressions, users can leverage their knowledge of mathematical notation in order to create mathematical sketches. Java applets that provide both interactive and dynamic illustrations have also been developed for exploring various mathematics and physics principles [CT98]. However, these applets are not general, typically provide limited control over the illustration, and rarely show the user the mathematics behind the illustration.

Alvarado [Alv00] and Kara [KGS04] let the user make sketched diagrams that are recognized as drawing primitives with domain knowledge from specific disciplines and then animated. Although these systems provide powerful illustrations of physics and mathematical concepts, they are limited because of their domain knowledge and because they hide the underlying mathematical formulations from the user. Pen-based systems have also been developed for other types of dynamic illustration. For example, Pickering et al. developed a system for sketching football plays, simulating them, and then creating a dynamic illustration of the play outcome [PBLP99] while Davis et al. developed a pen-based system for creating traditional animations [DACP04].

MathJournal, developed by xThink, Inc., is the closest in spirit to MathPad^{2}because its animation controls let users write down and recognize mathematics, make drawings, and assign the mathematics to the drawings. However, a key limitation of MathJournal's animation control is that users must keyframe their animations (typically providing a starting and ending frame), making the user interface less fluid and contravening how users would make diagrams with pencil and paper. In addition, MathJournal's animation control lacks the iteration and conditional constructs, diagram rectification, and modeless gestural user interface that mathematical sketching supports.

3. The MathPad^2 User Interface

To make mathematical sketches in MathPad^2, users write down mathematics, make drawings, and make associations between the two. Additionally, users can invoke mathematical tools such as graphing, function evaluation, and equation solving to help create and manipulate their sketches. In this section, we describe how users perform these tasks with MathPad^2's modeless gestural user interface. A summary of the commands are found in Figure 2.

When designing our modeless gestural interface, we wanted the gestures not to interfere with the entry of drawings or equations and still be direct and natural enough to feel fluid. To accomplish this, we use context sensitivity to determine what operations to perform with a single gesture. We also use the notion of punctuated gestures, compound gestures with one or more strokes and terminal punctuation, to help disambiguate gestures from mathematics and drawings. We also wanted to ensure that gestures which seem logical for more than one command should be used for all of those commands. For example, if a particular gesture makes sense for two or three different operations, then we want that gesture to invoke all those operations. More details on the design of and methodology behind these gestures can be found in [LZ04, LaV05].

To write mathematical expressions, users simply write them down using the stylus as if they were using pencil-and-paper. To have the system recognize a mathematical expression, users must lasso the expression and make a tap inside the lasso. Recognized symbols are presented to users in their own handwriting since MathPad^{2}has handwriting samples from individual users as a result of our writer-dependent mathematical expression recognition engine. When users move the stylus over the bounding box of the recognized mathematical expression, a green button appears in the box's lower right corner, and when pressed, shows whether the expression was parsed correctly. If a mathematical expression

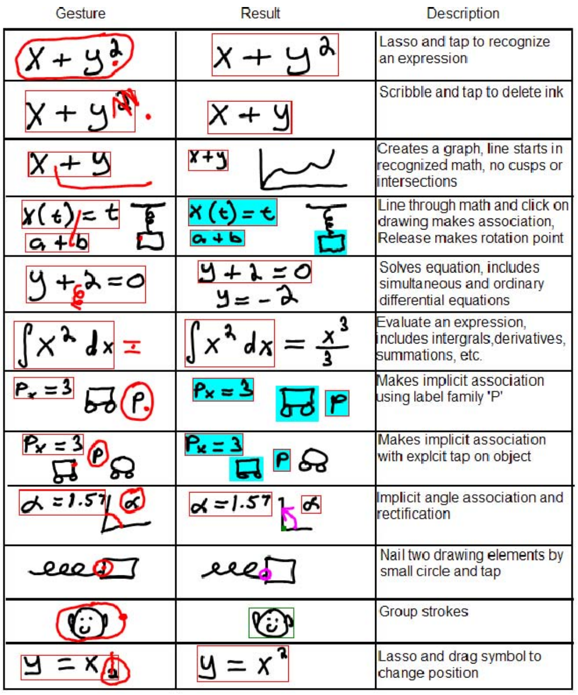

Figure 2: *MathPad^2's gestural commands. Gesture strokes in the first column are shown here in red. In the second column, cyan-highlighted strokes provide association feedback (the highlighting color changes each time a new association is made), and magenta strokes show nail and angle association/rectification feedback.*

is recognized incorrectly, users can simply erase the offending symbols using a scribble erase gesture followed by a tap and then re-recognize the expression. Users can also tap on a recognized symbol to get a list of alternates. If there is a parsing error with the mathematical expression, users can lasso the offending symbols and interactively move them to a new location where the complete expression will be reparsed.

Users make drawings in the same way they write mathematical expressions except that the ink strokes need not be recognized. We refer to these ink strokes as drawing elements and they can be grouped together to form composite drawing elements. Users lasso the drawing elements they want to composite and make a tap on the lasso line. Tapping on the lasso line distinguishes this operation from recognizing mathematical expressions. Users can also nail drawing elements together by drawing a small circle over them and making a tap inside the circle. Nailing drawing elements together lets users make stretchable objects. Note that the drawn circle must not completely contain any drawing elements in order to be recognized as a nail gesture. This constraint distinguishes it from the gesture for making composite drawing elements and recognizing mathematical expressions.

One of the most important components of MathPad^{2}is the ability to associate mathematics to drawing elements so they know how to behave during an animation. Users can make associations either explicitly or implicitly. Users make explicit associations by simply drawing a line through the bounding boxes of all the necessary mathematical expressions and tapping on a particular drawing element. As the stylus hovers over drawing elements, they highlight to give users feedback about which drawing element they will select. Implicit associations are made by labeling a drawing element with a variable name or constant value and can be either point or angle associations. Point associations are made in the same way that mathematical expressions are recognized except the tap is made on the drawing element instead of inside a lasso. Angle associations are made by drawing an angle arc and label. Then users lasso the label and make a tap whose location on the arc determines the *active line* — the line attached to the arc that will move when the angle changes. The apex of the angle is then marked with a green dot, and the active line is indicated with an arrowhead on the angle arc. In either case, MathPad^{2}uses the label to find all of the required mathematical expressions that should be associated to the drawing element.

Finally, MathPad^{2}provides users with a mathematical toolset for graphing and evaluating functions as well as solving equations that can assist users in making mathematical sketches. Users graph functions by simply drawing a sufficiently long, smooth line with no self-intersections, starting inside the bounding box of a recognized mathematical expression, intersecting any other functions along the way, and ending outside all expression bounding boxes. This gesture creates a graph control widget where users can view plots of the functions the graph gesture has intersected and also change the domain and range of the functions by writing down the values and pressing the update button.

Users evaluate mathematical expressions such as integrals, summations, and derivatives by writing an equal sign to the right of the expression and making a tap inside the equal sign's bounding box. The results are then displayed to the right of the drawn equal sign. Users solve single, simultaneous, or ordinary differential equations, by making a squiggly gesture (see Figure 2). This gesture is identical to the graphing gesture except the line must contain two self-intersections. The results are then displayed underneath the last intersected equation.

4. MathPad^2 Evaluation

4.1. Experimental Design and Tasks

The goal of our initial usability experiment is to get users' reactions to the prototype to validate the user interface design and its potential benefit as well as determine if further, more in-depth studies are needed. More specifically, we wanted to evaluate the intuitiveness of MathPad^2's user interface and gauge the perceived usefulness of the tool. Writing down mathematical expressions and making drawings is a fairly intuitive task, and although our gestural commands need to be taught, we felt they were designed so that they should be easy to understand given simple demonstrations of their use.

In the experiment, subjects must complete six tasks representing common interactions that a student or teacher would perform with MathPad^2. Before a subject performs each task, the experimenter shows the subject how to perform the required gestures for that task via demonstration only. Tasks 1–3 were designed to test how well users were able to use the graph, equation solving, and expression evaluation gestures. First, subjects are shown how to write and recognize mathematical expressions using the lasso and tap gesture, how to erase ink using the scribble erase gesture, and how to use the correction user interface. Then, they are shown how to perform each task specific gesture or command. For task 1 (Graphing), after being shown the required gestural commands, the subjects write, recognize, and then graph $y = x$, $y = x^3$, and $y = cos(x)e^x$. Then subjects change $y = x^3$ to $y = x^2$, graph the function, and change the function's domain from $-5...5$ to $0...8$. For task 2 (Equation Solving) task, subjects write down and recognize $x^2 - 16x + 13 = 0$ and solve the equation. Next, subjects write and recognize $x^2y + 2y = 4$ and $3x + y = 2$ and solve this set of simultaneous equations. For task 3 (Expression Evaluation), subjects write down the following expressions and evaluate them:

- $\int_0^2 x^2 dx$
- $y = \int x^2 cos(x) dx$
- $\frac{dy}{dx}$
- $\frac{d^2y}{dx^2}$
- $\Sigma_{l=0}^{5}(l-1)^2$.

In all tasks, subjects are instructed to use the correction user interface if the recognizer incorrectly recognizes symbols or expressions.

Tasks 4–6 were designed to lets users make mathematical sketches and evaluate whether they prefer to use implicit or explicit associations. Task five also was designed to evaluate how well subjects can make nails. Note that only task four required subjects to write down the necessary mathematical expressions. Tasks five and six used prewritten mathematical expressions because we felt having them write and recognize these expressions was not needed, given the many expressions they had already written in the mathematical expression recognition study (see Section 4.4). However, with task four, we wanted to see how well subjects could make a mathematical sketch from beginning to end.

The fourth task (Bouncing Ball), has subjects create a complete mathematical sketch of an object bouncing along the ground. Subjects write and recognize the four mathematical expressions shown in Figure 3, make a drawing with a horizontal line representing the ground and a composite drawing element consisting of three circles drawn near the start of the horizontal line. Next, subjects write the number 20 and associate it to the horizontal line. Finally, subjects associate the mathematics to the composite drawing element, either choosing an explicit association or using an implicit association with the letter "p" as a label, and run the sketch. Note that if MathPad^{2}fails to recognize subjects' mathematical expressions after several attempts, we provide them with prewritten expressions. However, we do not make them aware of this when the instructions for this task are given.

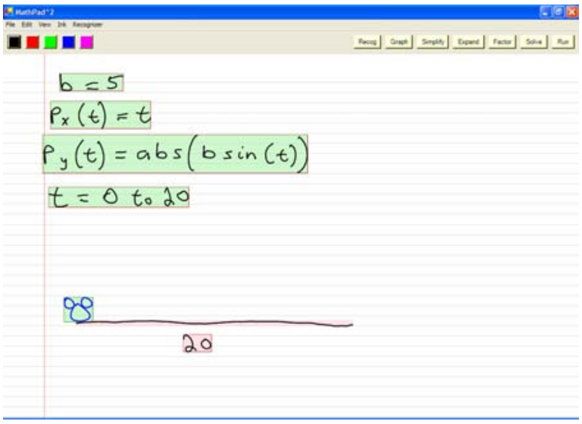

Figure 3: *The fourth task in the MathPad^{2}usability test.*

The fifth task (Oscillator) has subjects create a mathematical sketch illustrating damped harmonic oscillation. The experimenter instructs subjects to first draw a line and make seven nail gestures along that line. This subtask does not have anything to do with the mathematical sketch itself, but gives us additional accuracy data on how well subjects can perform the nail gesture. Subjects make a drawing consisting of a horizontal line, a spring underneath the line, and a box underneath the spring (see Figure 4). Subjects then use two nail gestures to nail the horizontal line to the spring and the spring to the box. Next, subjects associate the mathematics to the box, using an explicit or implicit association with the letter "y" as a label, and run the sketch.

In the last task (2D Motion), subjects create a mathematical sketch illustrating 2D projectile motion subject to air resistance (see Figure 1). Subjects draw a horizontal line and a ball near the left side of the horizontal line. They then associate the number 100 to the horizontal line. Finally, subjects associate the mathematics to the ball, using an explicit or implicit association with the letter "p" as a label, and run the sketch. After all six tasks are completed, subjects answer a post-questionnaire.

4.2. Participants

Seven subjects (four men and three women), participated in the MathPad^{2}usability evaluation. Subjects were recruited from the Brown University undergraduate population and

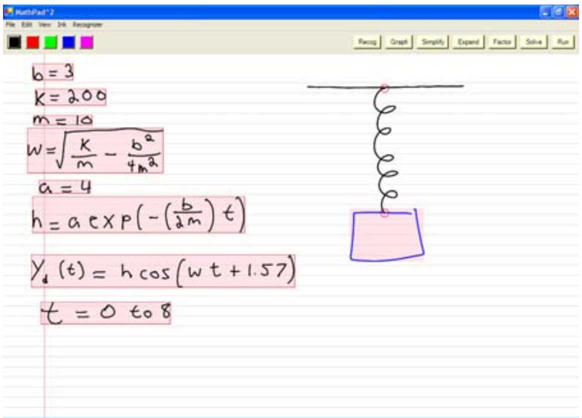

Figure 4: *Subjects create a damped harmonic oscillator in the fifth task.*

were either physics or applied mathematics majors. We chose this particular user population because MathPad^2 was designed for mathematics and physics students. Subjects' ages ranged from 19 to 23 and all were right-handed; only one had used a pen-based computer before (a PDA). All seven subjects were asked prior to the study if they had used mathematical software before and which packages: six subjects answering yes and had used a variety of different packages including Matlab, Mathematica, and Maple. All seven subjects were paid $30 for their time and effort.

4.3. Evaluation Measures

We evaluate MathPad^2's usability using quantitative and qualitative data from subjects' task performances and from a post-questionnaire. As subjects perform the six experimental tasks, the experimenter records important information about subjects' performances in completing each task, the decisions they made, and counts their mistakes. Performance is characterized by whether subjects can complete each task and how well they do on each subtask. Therefore, the experimenter records whether or not subjects make the appropriate gestures correctly and, if so, whether on the first attempt. Knowing how well subjects perform gestural operations on their first attempt is an important measure because it tells us how easy the gestures are to make and remember. The experimenter also records subjects' choices of implicit and explicit associations in tasks 4–6 so as to get a quantitative metric for their preferences.

After subjects have completed all six tasks they are given a post-questionnaire designed to get their reactions to the MathPad^2 user interface and its perceived usefulness as well as assess how well they remember certain gestures. The post-questionnaire consists of four parts. The first and second parts are adapted from Chin's Questionnaire for User Interface Satisfaction [CDN88] and asks subjects to rate MathPad^2's user interface as a whole and its individual components. The third part of the post-questionnaire, the recall test, asks subjects to show what gestures they would use for six different operations. The fourth part of the post-questionnaire was adapted from the Perceived Usefulness portion of Davis's questionnaire for user acceptance [Dav89] and asks whether subjects would use MathPad^2 in their work. After subjects answer the post-questionnaire, the experimenter reviews it with them to make sure their answers are clear and to elaborate further on any specific parts of MathPad^2.

4.4. Mathematical Expression Recognition

An important part of MathPad^2's user interface is that users can write down mathematical expressions as if they were using pencil and paper. Thus, mathematical expression recognition accuracy is an important part of the overall user experience. MathPad^2 uses a writer-dependent mathematical expression recognizer [LaV05] that includes a mathematical symbol recognizer and a mathematical expression parsing system. Each test subject had to provide handwriting samples to train the recognizer and this task took 50 minutes per subject. Note that subjects were given rest periods to ensure they did not get tired during training. Before completing the MathPad^2 tasks, we also had subjects write down symbols and a set of mathematical expressions to test the recognizer's accuracy Overall, the recognizer recognized symbols correctly 95.1% of the time with a standard deviation of 2.65%. The parsing component of our mathematical expression recognizer made correct parsing decisions 90.8% of the time with standard deviation of 4.47%. More detailed results on the mathematical expression recognition evaluation can be found in [LaV05].

4.5. Results and Discussion

4.5.1. Task Performance Results

For the first three tasks, subjects were able to write and recognize all of the mathematical expressions fairly easily. In some cases, they had to use the correction user interface to fix recognition errors, generally getting MathPad^2 to recognize their expressions on the second or third attempt. 27 out of 28 graphing operations (four per subject) were made on the first attempt. Subjects also had to change the domain of a graph; they all completed this operation on the first attempt. 12 out of 14 equation-solving operations (two per subject) were made on the first attempt. The other two equation solves were correctly performed on the second attempt. 34 out of 35 expression evaluations (five per subject) were made on the first attempt. One subject, however, did have difficulty in getting MathPad^2 to recognize $\frac{d^2y}{dx^2}$ and even after multiple attempts was not able to evaluate the expression.

All seven subjects were able to complete tasks 4–6 making the dynamic illustrations. Subjects also had no difficulty in making the drawings for each task and only once did a subject have trouble making a composite drawing element. In the Bouncing Ball task, 12 out of 14 associations were made on the first attempt and 8 of them were done implicitly. Three subjects did have difficulty in getting MathPad^{2}to recognize the required mathematical specification for the Bouncing Ball task and, after multiple attempts (about 10 minutes), were given prewritten expressions. The difficulty was not in symbol recognition, but in expression parsing. Two of these subjects had parsing decision accuracies below 90% in the mathematical expression test while the other subject's accuracy was 92%. This result provides evidence indicating that higher parsing decision accuracy is needed. In the Spring task, 56 out of 63 nails (seven per subject) were made on the first attempt. Most of the remaining nails were made on the second attempt. However, one subject required several attempts to make the necessary nails and had to recreate the drawing after inadvertently erasing part of it when erasing an incorrectly recognized nail. Subjects had to make one association in this task, and all seven were made on the first attempt explicitly. For the 2D motion task, subjects made 12 out of 14 associations on the first attempt with all of them made implicitly. One subject did had some difficulty with the implicit associations and needed several attempts to make them correctly.

Overall, subjects did well on all six tasks, considering they had no hands-on training beforehand. Their first attempt performances are summarized in Table 1. Subjects hand no difficulty in making a lasso and tap to recognize mathematical expressions or in using the scribble erase gesture. In only one case did a subject not complete part of a task and this was due to MathPad^2's inability to recognize an expression correctly. Subjects made 160 out of the 175 gestural operations correctly (91.4%) on their first attempt. This number is high considering that subjects had not practiced any of the gestural commands. One subject did have some difficulty with implicit associations due to problems with making taps. The greatest problem subjects had with the six tasks was obtaining correctly recognized expressions in certain situations. That three out of the seven subjects required prewritten mathematics for the Bouncing Ball task shows that the mathematical expression recognizer needs improvement.

4.5.2. Post-Questionnaire Results

Overall Reaction. Table 2 summarizes subject's overall reaction to MathPad^{2}and shows that they had a positive reaction to the prototype. When subjects were asked why they chose their rankings, most asserted that MathPad^{2}works well, is easy to use, and would be very useful for students in a classroom setting and/or doing homework problems. One subject was "amazed at the application's power". Two subjects claimed MathPad^{2}was easy to use but could be frustrating when it had trouble recognizing their handwriting; this frustration explains why the second and third rankings in Table 2 are slightly below the first and fourth rankings.

Ease of Use. Subjects rated different parts of the MathPad^{2}user interface from 1 (easy) to 7 (hard). Table 3 summarizes these results and shows that subjects found the tasks they had to perform easy to do. Subjects gave recognizing expressions the highest average ranking, indicating the fact that some users had trouble getting MathPad^{2}to recognize their handwriting. When asked about their ranking, they stated that the gesture for recognizing mathematical expressions (i.e., lasso and tap) was easy to do, but the results of the recognition operation led them to choose a higher ranking on the easy (1) to hard (7) scale.

First Attempt Gesture Performance Summary			
	Completed	Total	Percentage
Graphing:	27	28	96%
Equation Solving:	12	14	86%
Exp. Evaluation:	34	35	97%
Nails:	56	63	88%
Associations:	31	35	89%
Total:	160	175	91.4%

Table 1: *A breakdown of test subjects' first attempt gesture performance.*

Overall Reaction to MathPad^2		
	Mean	Std. Deviation
Terrible=1, Wonderful = 7	6.42	0.54
Difficult=1, Easy=7	5.57	0.98
Frustrating=1, Satisfying=7	5.57	1.13
Dull=1, Stimulating=7	6.14	0.38

Table 2: *Subjects' average ratings of their overall reaction to MathPad^{2}on a scale from 1 to 7.*

MathPad^{2}User Interface Ease of Use		
	Mean	Std. Deviation
Writing Mathematics	1.43	0.97
Recognizing Mathematics	2.57	1.81
Graphing Functions	1.0	0.0
Solving Equations	1.0	0.0
Evaluating Expressions	1.0	0.0
Grouping Drawing Elements	1.57	0.79
Making Associations	1.71	0.76
Making Nails	1.57	0.59

Table 3: *Subjects' average ratings of ease of use for different components of the MathPad^{2}user interface (scale: 1=easy, 7=hard).*

Association Preference. All seven subjects preferred explicit associations, claiming they were easier to remember and simpler and faster to perform. However, they did say that when associations need to be made with a drawing element and a large set of mathematical expressions, the implicit method is more appropriate. We can thus conclude that both association methods have their place in mathematical sketching.

Correction User Interface. Five out of the seven subjects tested found the correction user interface helped them. The two subjects who said no claimed that the alternate lists gave them no help in correcting recognition errors. One subject wanted more choices to appear in the alternate lists, especially in the equation alternate list.

Positive and Negative UI Aspects. Most subjects identified the most positive aspect as its ability to quickly make drawings move as described by mathematical equations. Two subjects claimed that solving equations was one of the user interface's most positive aspect. One subject thought that the best part of MathPad^2's user interface was the scribble erase command; another subject said the user interface's simplicity was its most positive aspect. Three subjects stated that getting MathPad^{2}to recognize certain symbols and equations correctly was the most negative aspect of the user interface. Two subjects stated that the lack of interactive feedback for implicit associations was a significant drawback, and one subject stated that a negative aspect was the time necessary to get used to the gestural commands. Finally, two subjects said that MathPad^2's user interface had no negative aspects.

Overall Ease of Use. On average, subjects gave MathPad^{2}a 1.86 (1 equals easy and 7 equals hard) with a standard deviation of 0.69. When they were asked to explain their ratings, two dominant themes emerged. First, subjects found the interface easy to use and remember, but were in some cases frustrated by problems in mathematical expression recognition. However, the subjects who had trouble with recognition all felt it would improve with more practice. Those subjects were also asked if they would still use MathPad^{2}in spite of their recognition problems; they all said they could deal with these problems because of the functionality MathPad^{2}would give them. Second, subjects felt the interface was easy to use once it was explained, a result that helps to validate our demonstration-based teaching protocol.

Gesture Recall Test. Subject were asked how to invoke gestural commands for graphing, solving equations, evaluating expressions, recognizing a mathematical expression, making nails, and making implicit associations. This part of the questionnaire took place about 5 to 10 minutes after they used MathPad^2. Subjects answered 38 out of the 42 recall questions correctly (six per subject) for a recall rate of 90%. Of the four questions subjects answered incorrectly, three subjects missed the equation solving gesture (squiggle) and one missed the expression evaluation gesture (equal and tap). The 90% recall rate indicates that subjects had little difficulty remembering MathPad^{2}gestures except for the equation solving gesture. Even though three out of the seven subjects forgot the equation solving gesture, they still claimed it was easy to use based on their mean ranking in Table 3.

Likely Usage. Table 4 summarizes subjects' ratings on the different "perceived usefulness" statements, on a scale of 1 (unlikely) to 7 (likely). Most subjects would use MathPad^{2}in their work. When asked to explain their ratings, four subjects stated that the application would help them to do their classwork and obtain a better understanding of problems and concepts. However, there was no consensus on whether MathPad^{2}would speed their understanding of these problems and concepts. One subject said that the ability to quickly solve equations and make graphs would be very beneficial. Two subjects said they did not think they would use MathPad^{2}in its current form in their work (explaining the high standard deviations in Table 4). Both of these subjects work in theoretical physics, one in optics and the other in modern physics. However, one of these subject stated she would have used MathPad^{2}during beginning physics classes while the other stated he would use MathPad^{2}if it had support for light ray and optics diagrams. Finally, all seven subjects felt the application would be a good tool for teachers of introductory mathematics and physics classes.

MathPad^{2}Perceived Usefulness		
	Mean	Std. Deviation
Accomplish Tasks Faster	5.14	1.95
Improve Performance	4.71	2.36
Increase Productivity	5.0	1.91
Enhance Effectiveness	5.14	2.04
Easier To Do Work	5.57	1.90
Useful In Work	5.42	2.37

Table 4: *Subjects' average ratings of the perceived usefulness of MathPad^{2}in their work (scale: 1=unlikely, 7=likely).*

4.5.3. Discussion

The results of our initial MathPad^{2}usability study suggest that, based on our evaluation criteria, the MathPad^{2}user interface is, in general, intuitive with subjects picking up the interface with relative ease. With only minimal training, most gestures are easy to remember and use. However, if we examine the first attempt task performance results (Table 1) in conjunction with the recall test from our post-questionnaire, we see that the equation solving gesture has the lowest first attempt accuracy and was the most difficult to remember. This indicates that this gesture is not as intuitive as the others. Additionally, if we look deeper into users' preferences for making associations, we see that they preferred explicit associations and of the four associations that were not made on their first attempt, all four were implicit. Again, this result suggests that explicit associations are more intuitive than implicit ones. First attempt performance for making nails was also a bit lower than expected, but we feel this might have been an implementation issue. In terms of perceived utility, subjects think the application is a powerful tool that beginning physics and mathematics students could use to help solve problems and better understand scientific concepts.

Most subjects performed the tasks with little trouble, while a few had some difficulty, stemming primarily from problems with mathematical expression recognition. However, these subjects also said they were willing to accept these recognition problems, given what MathPad^{2}can offer them. This result is somewhat contrary to our expectations about the negative impact of our mathematical expression recognizer on MathPad^{2}usability. Nevertheless, we need better mathematical expression recognition that will perform robustly across a larger user population. Although these results do not tell us how much more accurate the recognizer needs to be, its clear that a mean accuracy of 90.8% for making correct parsing decisions is too low. A better correction user interface could also go a long way to helping with users' frustrations when incorrect recognitions occur. In addition, more interactive feedback is needed for implicit associations, and the equation solving gesture should be redesigned.

Although the results of our initial evaluation are positive, we recognize it can be argued that there are two limitations with our study. First, we only used seven test subjects. We could have had more subjects, but we felt that seven was appropriate for an initial evaluation of MathPad^{2}and its gestural interface, given one of our main goals was to determine whether larger studies were needed. Second, we did not compare MathPad^2's user interface with any other interface metaphors. Although this could be considered a limitation, our goal in this evaluation was to determine how well users could use the MathPad^{2}interface, not whether it was better than any other interface. For this work, we feel our experimental design was suited to answering our intended questions. However, as we perform future usability tests to gain a deeper understanding of the benefits of mathematical sketching, we will need more comparative experimental designs with larger subject numbers.

Given the results of our evaluation, we plan to make improvements to MathPad^{2}by adding more functionality and improving the weaker points of the interface as well as improving the parsing component of our mathematical expression recognizer. Given the generally positive results of our evaluation, we are confident in pursuing further MathPad^{2}experimentation. Thus, we plan to explore the pedagogical benefits of MathPad^{2}in a summative evaluation where students will use MathPad^{2}as part of a mathematics or introductory physics course.

5. Conclusion

We have presented an initial evaluation of MathPad^2, a prototype application for making dynamic illustrations using the mathematical sketching paradigm, to test its intuitiveness and perceived utility. Our evaluation suggests that MathPad^2's user interface is generally intuitive, although some parts of the interface need to be reevaluated. Additionally, the MathPad^{2}application is perceived to be a powerful tool for exploring mathematics and physics concepts. Although some of our test subjects had some difficulty with getting the system to recognize their mathematical expressions, they still gave MathPad^{2}positive feedback and would use MathPad^{2}regardless of these issues because of its functionality. These results also support future MathPad^{2}development and longer term evaluations.

Acknowledgements

Special thanks to Robert Zeleznik and Andries van Dam for valuable discussions. This work is sponsored in part by a gift from Microsoft.

References

[Alv00] ALVARADO C.: *A Natural Sketching Environment: Bringing the Computer into Early Stages of Mechanical Design*. Tech. rep., Master's Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 2000. 2

[Bor79] BORNING A.: *A Constraint-Oriented Simulation Laboratory*. PhD thesis, Stanford University, 1979. 2

[CDN88] CHIN J., DIEHL V. A., NORMAN K. L.: Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the ACM Conference on Human Factors and Computing Systems (CHI'88)* (1988), pp. 213–218. 1, 5

[CT98] CHRISTIAN W., TITUS. A.: Developing web-based curricula using java physlets. *Computers in Physics 12*, 3 (May-June 1998), 227–232. 2

[DACP04] DAVIS J., AGRAWALA M., CHUANG E., POPOVIC Z.: A sketching interface for articulated figure animation. In *Proceedings of the Eurographics/SIGGRAPH Symposium on Computer Animation* (2004), pp. 320–328. 2

[Dav89] DAVIS F. D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly 13*, 3 (Sept. 1989), 319–340. 1, 5

[KGS04] KARA L. B., GENNARI L., STAHOVICH T. F.: A sketch-based interface for the design and analysis of simple vibratory mechanical systems. In *Proceedings of ASME International Design Engineering Technical Conferences* (2004). 2

[LaV05] LAVIOLA J.: *Mathematical Sketching: A New Approach to Creating and Exploring Dynamic Illustrations*. PhD thesis, Brown University, 2005. 1, 2, 5

[LZ04] LAVIOLA J., ZELEZNIK R.: Mathpad^2: A system for the creation and exploration of mathematical sketches. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004) 23*, 3 (August 2004), 432–440. 1, 2

[PBLP99] PICKERING J., BHUPHAIBOOL D., LAVIOLA J., POLLARD N.: *The Coach's Playbook*. Tech. rep., Master's Thesis, Department of Computer Science, Brown University, CS-99-08, May 1999. 2