Designing a Collaborative Finger Painting Application for Children

Hilary Browne, Ben Bederson, Allison Druin, Lisa Sherman
Human-Computer Interaction Lab
Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742
{hbrowne, bederson, lsherman}@cs.umd.edu, allisond@umiacs.umd.edu

Wayne Westerman
Department of Electrical and Computer Engineering
University of Delaware
Newark, DE 19716
westerma@ee.udel.edu

ABSTRACT
We describe the design and implementation of a collaborative, computer-based finger painting program for children using a new hardware input device called a Multi-Touch Surface (MTS). The MTS uses a flat surface about the size of a keyboard to track multiple, simultaneous finger motions, which we transform into paint strokes on a screen. We describe related work and explain how our program design was guided by the suggestions of children. We discuss the hardware and software of the MTS and the challenges of designing our program. Finally, we present the Finger Painting Table, a collaborative, embedded application built using the MTS, and discuss future work.

Keywords
Multi-Touch Surface, Finger Painting, Children, Computer Supported Cooperative Work (CSCW), Educational Application, Single Display Groupware (SDG)

INTRODUCTION
The motivation for this project came out of the ongoing research goal in our lab of designing technologies that combine "the power of computation with the familiarity of a child's world" [1]. In particular, we are working to create a kindergarten "Classroom of the Future" that supports natural interaction and collaboration among children using embedded technologies. In this paper, we discuss our work designing and implementing one such technology, a collaborative finger painting program.

We used a new, pre-prototype hardware device called a Multi-Touch Surface (MTS) [12] as an input device for creating computer-generated paint strokes. We then built a Finger Painting Table by embedding the surface in a soft, colorfully decorated table with a large projection surface for showing the results of individual or collaborative painting activities. The MTS is a gesture-based input device that can sense and track the motion of multiple fingers on a curved, rectangular surface of about 20x8 inches (51x20 cm). Hardware sensors and special software allow the surface to sense the location of multiple fingers and differentiate gestures including typing, pointing, and clicking. While the typing and mouse gesture capabilities of the surface are impressive, we were mainly interested in the finger tracking functionality for creating a program that would allow young children to paint using just their fingers on the surface. Using the MTS and its software, finger paths on the surface can be transformed into brush strokes of color in a window on a computer screen (Figure 1).

Figure 1: The Multi-Touch Surface and attached monitor. Finger strokes made on the board are transformed into paint strokes on the screen.

Computer-based painting programs have been around for many years, but these all require the use of an input device such as a mouse or stylus to control both painting and selecting paint options such as colors or patterns. Such devices can be tricky and unnatural for small children to use, particularly if many steps or separate keyboard actions are required to select different painting options. Using the MTS as an input device instead offers four distinct advantages over other devices:

1. It is easy to use – it doesn't have to be picked up, moved, or manipulated.

2. It acts as multiple devices because it can sense all 10 fingers at once – each finger can potentially do something different, such as paint a different color.
3. It can be used both for painting and for option selection – it replaces both mouse and keyboard.
4. It supports collaboration with multiple users – two or three children can work together simultaneously without fighting over the device.

RELATED WORK

A few researchers such as Lee at the University of Toronto [20, 21] and reportedly R. Boie and L. Nakatani at Bell Laboratories (see [22] for early review) built multiple-touch-sensitive devices in the mid-1980's, but applications were not forthcoming at that time. FingerWorks' Multi-Touch Surface, developed by Elias and Westerman at the University of Delaware [12] following commercial success for single-finger touch pads, is the first multi-touch device to provide pointing, typing, and chord gesture recognition in addition to advanced finger tracking algorithms. Because FingerWorks' sensing technology scales to arbitrary surface sizes and resolutions, the MTS is suitable for a wide range of new interaction studies ranging from finger painting to defense command centers. Other work in six different areas is also related to our finger painting project.

First, the design of children's software using familiar objects in a child's world, particularly soft, pleasant to touch things like stuffed animals, has become increasingly important in introducing computers to young children. Research projects such as the MIT Media Lab's SAGE [26] and the University of Maryland's PETS [9] use stuffed, robotic animals with embedded computers to allow children to create and tell stories. Commercial products such as Microsoft's Actimates Barney [25] have unleashed a new generation of interactive stuffed animals. Our goal in creating an embedded application with the MTS was to create the same seamless integration between computer hardware and a soft, pleasant painting environment.

Second, research in the design of large, physical interactive spaces for children is related to our goal of integrating an embedded finger painting application into a kindergarten classroom of the future. Work in designing physical interactive spaces for children such as NYU's Immersive Environments project [10] and MIT's KidsRoom [4] has focused on using state of the art technologies and construction to create interactive, immersive, user-controlled experiences. Work at the University of Maryland on StoryRooms [1] has focused on lower cost methods of creating the same kinds of interactive experiences to allow children to create physical storytelling experiences. Our design of the Finger Painting Table was motivated by similar low cost materials and techniques because of the likely budget constraints of a kindergarten classroom.

The third area of related work involves research in collaborative technologies and environments. Our initial evaluation (see *Informal Testing* section) indicates that the MTS is very conducive to shared use. The surface is large enough to accommodate multiple hands, and each finger on each hand can do something different. However, most recent research in the area of collaborative input devices has focused on using multiple devices such as mice, rather than a single device like the MTS. The use of multiple mice has been shown to influence children's learning and behavior in collaborative activities [16] and to improve and enhance collaboration [2, 23]. Although the MTS is a single device, it recognizes multiple finger inputs, so we anticipate many of the same benefits will apply.

The fourth type of related research concerns the hardware used to identify and track fingers on the MTS. Similar devices include the commercially available tablet and stylus used by art programs and the touch pad standard on many laptop computers. However, the MTS technology is more advanced because it tracks multiple devices simultaneously, can directly detect fingers, and has some ability to differentiate between them. The only other technologies that achieve these types of advanced tracking capabilities make use of computer vision techniques, some of which we hope to explore and compare to the MTS in the future.

One of the earliest vision-based hand tracking devices was the VideoDesk, built by Krueger in 1987 [19]. It consisted of a light table with a camera mounted above it that identified and tracked users hands. The silhouette image of the hands appeared on a monitor and could be used to perform various input activities. A similar design was used at Xerox to create the DigitalDesk, which combined document and pointing device recognition [28]. At Stanford, researchers are currently attempting to track and distinguish multiple laser pointers with cameras for use on a large, high-resolution display called an interactive mural [30]. In [7], the authors describe a technique for glove-free tracking of hand movements in three dimensions. In MIT's KidsRoom project [4], the authors use context-sensitive, remote sensing to track people and motions depending on the application being used without the need for sensors embedded in gloves, head displays, microphones, etc. Intel's Me2Cam [17] allows children to control computer games with a camera mounted on a computer that tracks their motions and gestures when they stand in front of it.

Some of these techniques could prove useful in creating a finger painting program separate from or in conjunction with our own, but to date they have not been used. In particular, a camera located above a surface such as the VideoDesk could provide perfect finger identification under most circumstances, something the MTS cannot always do. However, a camera would have a difficult time sensing exactly when the fingers were touching the surface and how much pressure was being applied. Cameras may not be able to report fingertip position as precisely or accurately either. Thus, a combination of MTS and vision hardware may provide the best solution for a finger painting application.

The fifth area of related work involves computer-generated brushes for drawing and painting programs. A number of previously developed techniques were useful in helping us understand how to efficiently create realistic brush strokes. The most common technique for generating paint strokes on a screen is to transfer a pre-defined image along a stroke path defined by a user, either free-hand or with control points [3, 15, 29]. In [3], the authors describe a charcoal sketching system where users can control the location, pressure, and tilt of a charcoal brush using a stylus and tablet. Values obtained from the tablet are rounded to correspond to an image in a discrete range of predefined images of charcoal strokes. The entire image is then transferred to the location, rather than just painting a single pixel. We used a similar approach for our program, predefining a palette of brushes and accessing them according to discrete values for size, pressure, and color.

A number of researchers have explored a more complex model for computer-based painting by simulating calligraphy brushes [6, 18, 24]. Individual brush bristles, ink absorption and diffusion, and brush angle can all be modeled to dynamically vary different strokes. In [27], the authors experimented with six different position and orientation parameters for defining a brush. We did not model all of these details in our implementation, but may in the future. We chose to allow the diameter of brush strokes to change as a user touches a larger area of the MTS or touches the same area for a longer period of time, simulating a paint blot.

Finally, the sixth area of related research involves one of the most challenging aspects of designing computer-based painting programs that use touch sensitive input devices: allowing users to control where and how paint appears on the output device. In [5], the authors discuss these two issues as they relate to a touch sensitive tablet. Unlike a program that uses a mouse for input, there is no cursor to indicate where paint will show up when you touch the tablet. For the MTS, this is even more complicated because there can be up to 10 areas where paint will show up. Our solution to letting users know where paint would show up was to draw temporary cursor marks on the screen when users touched the MTS lightly. Pressing harder would cause painting at that location.

In [5], the authors also noted that using a tablet for both painting and controlling brush properties could be difficult and confusing. They suggested laying templates over the tablet to divide it into areas for drawing and areas for control features. The MTS has the ability to recognize different kinds of finger combination gestures for control versus painting, but we believe that these gestures would be too complicated and difficult to remember for small children. We may try using the template idea in the future. Currently, control functions such as changing brush properties and clearing the screen in our program require the use of a mouse. However, this is a temporary solution that we do not believe is appropriate for young children.

Figure 2: Members of our intergenerational design team work on designing the "Classroom of the Future".

DESIGN METHODOLOGY AND MOTIVATIONS

Before our MTS actually arrived and before we began designing our finger painting program, we described the MTS to six children. We asked them how they might want to paint with it and what features a painting program that used it should have. These children were between the ages of 6 and 11 and all had been members of the intergenerational design team in our lab for at least a year (Figure 2). The children come in twice a week after school and work with adults to design and test new technology for children [8]. Although the description of the MTS was rather abstract for the children, we received at least three interesting and useful design suggestions.

First, at least one child suggested that painting should be controlled with modes, rather than tools. She clearly understood that there would be no need for clicking on a tool such as a paintbrush and then dragging it around to paint. Rather, she suggested placing color swatches and brush shapes around the perimeter of the screen or the MTS that could be touched with a finger to set the properties of that finger for painting. Unfortunately, the MTS sensors are not powerful enough to differentiate between particular fingers in all situations, but the idea of using modes to assign properties to fingers is one we used.

Second, a number of children wanted various kinds of "accessories" to go with the finger painting program, including background colors or pictures, a library of images to use depending on the selected background, physical shapes to draw with instead of fingers, and sounds. Backgrounds, images, and sounds are all feasible, and we plan to implement them in the future. Painting with physical tools is possible if the tools are made of or encased in a conductive material (such as aluminum foil). However, the current implementation of the MTS is not designed to recognize different shapes, so we did not pursue this idea. In the design of the Finger Painting Table, we did use accessories in the form of colored shapes and objects to augment paintings projected onto the table (see *Finger Painting Table* section).

Third, some children suggested using different hand and finger gestures for different actions and controls. For instance, they suggested that dominant and non-dominant hands could have different responsibilities or the thumb could be used for special tasks such as mixing colors. Currently, we don't plan to implement these features because we feel that the gestures might be difficult for small children to remember. However, we do plan to explore using the surface or other custom input devices for different control activities, rather than our current setup, which requires the use of a mouse to perform control functions. All of these brainstorming ideas paved the way to establishing a general direction for our research. In the sections that follow, we discuss the MTS technology, design challenges, and the subsequent design iterations for our finger painting application.

MTS TECHNOLOGY

The MTS consists of a flat surface mounted over a grid of sensor chips (Figure 3). FingerWorks' MultiTouch technology includes the sensor hardware and low-level software for sensing, tracking, and recognizing hand and finger motion on the surface. The surface is curved in an arc shape for ergonomic comfort when typing. A serial cable plugs into a PC serial port to send finger-tracking data to the software. PS/2 mouse and keyboard cables can also be plugged into their respective ports on a PC to use the mouse and keyboard recognition functionalities of the surface. The MTS arrived with a laminated overlay of a keyboard for use when typing, but we removed this and replaced it with a plain paper covering for finger painting and did not use the keyboard and mouse cables.

FingerWorks' GestureScan software comprises low and high-level code for processing the input from the surface and makes it accessible for application programmers in Java. The software can report touch activity on many levels, from raw surface proximity images to identified finger trajectories to wholly recognized typing, pointing, and multi-finger gestures. We received the most advanced version, but only make use of fingertip shapes and trajectories. This information is made available to application programmers via a Java package called MID [14]. MID (Multiple Input Devices) was designed at the University of Maryland and supports input from multiple devices, in this case multiple fingers. The GestureScan software implements MID interfaces and native code for processing MTS input, and application programmers implement a listener interface for handling finger events. The events provide an array of the most recent finger contacts, which contain such information as their location, orientation, time of contact, and probable finger (i.e. left index finger). This information can then be used to create

Figure 3: The MTS senses finger position, pressure, and various gestures.

paint marks in a window on screen using standard Java graphics methods. More information about the MTS can be found in [12].

DESIGN CHALLENGES

There were a number of interesting design issues involved in creating a finger painting application for young children. The first had to do with creating realistic and fun brushes to paint with. The MTS came with a simple finger painting program that painted ellipses for each finger contact according to the finger location and pressure. However, there were a few problems with this strategy. First, due to the MTS's limited imaging frame rate, fingers can appear to jump a few centimeters between frames during rapid drags across the surface, leaving a trail of sometimes well-spaced ellipses rather than a smooth line. To solve this problem, we implemented a modified version of the standard Bresenham line-filling algorithm [13] to fill in gaps when samples from a finger path were far apart (Figure 4).

A second problem was that creating strokes by filling ellipses did not provide enough flexibility for creating brush strokes with different colors, patterns, and shapes. To address this problem, we predefined a set of our own brush images of various sizes and colors. We used these images to paint each brush mark instead of filling ellipses. This allowed us the flexibility to control the color and transparency of every pixel in every brush image. Finally, the paint strokes did not leave larger and larger blot marks when fingers were held in the same place on the MTS for a long period of time, as real paint might. To solve this problem, we tracked the time, position, and last brush mark size of each finger contact and painted larger marks of paint for contacts that did not move over a period of time.

Another important issue involved the assignment of colors for fingers when painting. The finger painting program that came with the surface pre-assigned a different color to each of the 10 fingers. We liked this idea, but the implementation had a serious problem. The color assignments depended on the surface being able to recognize exactly which fingers were being used. This could only be accomplished if fingers were put down in their "home" locations on the surface or if five fingers from one hand were put down simultaneously. If you only put down one finger to paint, the surface would guess which finger it was based on its location on the surface. For instance, if you put your right thumb down in an area where the left pinky would normally be placed, the left pinky color would show up, rather than the right thumb color. Although it was nice to be able to associate a particular color with each finger, the implementation was too unpredictable and restricting if you wanted to paint with a particular color in a certain area, a scenario we assumed would be fairly common.

We designed a different color assignment strategy to avoid this problem. Colors are assigned according to the temporal order in which fingers are placed on the MTS. The default settings include 10 colors, always assigned in the same order. Thus, the first finger that touches the surface paints red, no matter which finger it is or where on the surface it touches. The second finger paints orange, and so on. Changing the order of the colors, the type of brush, and other issues of control such as clearing the screen are currently done with menus and buttons using a mouse. Currently, we have 5 different brushes and 10 different colors. All fingers use the same brush type, and the order of the 10 colors can be permuted in a fixed manner. However, these restrictions are purely for simplicity and could be relaxed. In the future, we would prefer to enable children to make such control changes using the MTS alone.

A final design challenge involved the issue of indicating where paint would show up on the screen when a user touched the MTS. The finger painting software that came with the MTS had no provision for doing this, which was frustrating if one wanted to paint with more control. We attempted to solve this problem by setting a pressure threshold for painting. Touching the MTS with a light pressure results in a cursor mark being painted on the screen for a brief period of time and then disappearing. Touching the surface with a stronger pressure paints as usual. We added a menu option to enable and disable the cursors in case users found them distracting or just wanted to paint without them.

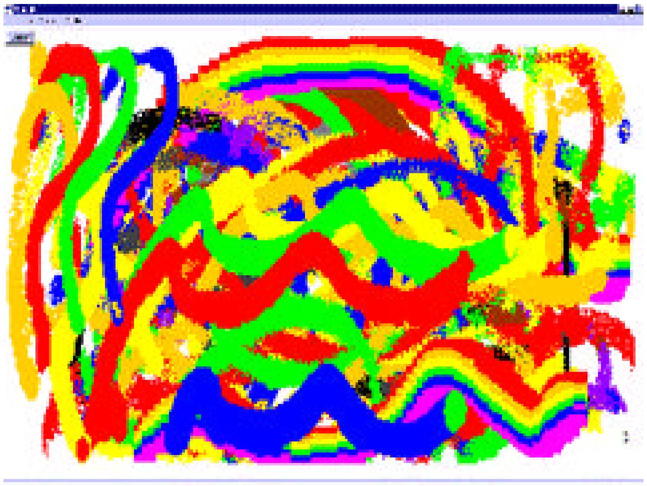

Figure 4: A painting created with the MTS. Smooth strokes with various colors and brush shapes are shown.

INFORMAL TESTING

After we completed the initial design of our program, we did some informal testing with 7 children, 4 of whom had participated in our initial session of pre-design questioning. The remainder of the children were new members of our lab design team. The children used the MTS with the painting program in groups of 2 or 3 for about 10 minutes for each group. We gave the groups very little instruction, essentially just letting them sit down and discover how things worked. While the groups worked, two adults observed them and took notes using the method of contextual inquiry, as described in [8]. One observer recorded the children's activities and the other recorded what they said. Both noted the time of the actions and the notes were synched up later.

The most interesting and encouraging thing about the testing sessions was that all of the children immediately liked and understood how to use the MTS. Four children specifically said it was "cool," and most groups didn't want to stop when their time was up. The children also shared the MTS remarkably well, dividing the surface up equally according to where they were sitting. Teams of 2 seemed to work particularly well, but teams of 3 seemed cramped. Using the surface for collaborative tasks and projects thus seems to be a feasible idea.

Most of the children just scribbled with their fingers and enjoyed seeing the different colors fill the screen. However, some children indicated that they didn't like how quickly the screen filled up, and only one child attempted to draw anything controlled – he wrote his name. This suggested to us that we should make the drawing window larger, and/or make the paint marks thinner to allow for more space and finer control. None of the children seemed to use the cursor feature, nearly always pressing hard enough to generate paint marks. More study is needed to determine if the cursor feature would be more useful if the children were trying to draw in a more controlled way.

Initially, none of the children used the mouse to select new brush colors and shapes or clear the screen. Most groups asked how to clear the screen and had to be shown the Clear button. The observers also had to point out the brush menus to encourage the children to try them out. As anticipated, these control functions were less than ideal. The children sometimes fought over control of the mouse, and some children wanted more control over picking colors. However, the children enjoyed being able to paint with the different brushes and colors. This suggests that we are on the right track by providing different brushes and

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.