The Geometer's Workbench:
An Experiment in Interacting with a Large, High Resolution Display

François Guimbretière, *Terry Winograd*
Computer Science Department
Stanford University
Stanford, CA 94305-9035
{francois,winograd}@cs.stanford.edu

Sha Xin Wei
Department Of Mathematics
Stanford University
Stanford, CA 94305-2125
xinwei@stanford.edu

ABSTRACT

We describe the Geometer's Workbench, a graphical front end for Mathematica running on the Interactive Mural, a large high resolution display. Our system bridges the gap between casual whiteboard interaction and the more formal use of a tool like Mathematica. We anticipate that the experience gained from designing and testing such a tool can be generalized to a large class of technical and scientific applications that use large, high resolution displays.

KEYWORDS: Interactive display, geometry, Mathematica, whiteboard, Ebeam

INTRODUCTION

In the past few years, large wall-based high-resolution displays have been developed in a number of research projects [28, 24, 7, 14, 1] Several graphics projects have explored scalable designs based on commodity hardware [28, 24, 7, 14], and we can expect that within a few years, displays of this type will be widely available and economically practical. So far most of the effort has focused on the mechanical and optical complexities of large tiled displays and on software that allows parallel rendering engines to reach a level of performance similar to machines an order of magnitude more expensive. At the same time, new tracking technologies, such as video camera tracking of a laser pointer [5] and ultrasonic tracking of markers [10, 15], have made it possible to interact directly on large screens with pen-like devices. But so far little work has been done to explore how the specific affordances of such a display can be leveraged in designing human-computer interfaces. This paper presents an initial experiment in that direction.

We fitted the Stanford Interactive Mural [14]—a large high resolution (60 dpi) display—with an EBeam pen tracker and used MathLink to connect our rendering engine to Mathematica, a powerful symbolic computation system for mathematics. The resulting system, which we call "the Geometer's Workbench," is a tool for mathematicians exploring or teaching algebraic geometry.

Even though the Geometer's Workbench is a specialized application. we anticipate that the experience we have gathered while building this application can be generalized to a wide variety of technical and scientific areas. The main features of the Workbench fall in 4 categories:

- **Informal interaction for scientific exploration tools** A geometer working at a blackboard employs a mixture of sketching, algebraic calculation and logical inferences, switching easily between these different modes of work on the same surface, taking advantage of a large surface to array and organize the written marks. A principal goal of our work was to make the power of a symbolic algebra engine like Mathematica available on a drawing surface that appears seamless to the user.
  Since our system is intended for casual multi-person shared use of an interactive surface with a large display area, standard mechanisms such as menu bars and modal dialog boxes are generally unsuitable. We developed the FlowMenu [11], a new style of interaction that is centered on a selection point on the board, and that smoothly integrates noun selection, verb selection and parameter entry. It requires no persistent "widgets" on the screen, leaving an uncluttered space looking more like a blackboard than a GUI desktop.
- **High resolution affordance** Our display has more pixels than a page printed at 300dpi with a .75" margin, but is dynamic, with substantial rendering power. This gives us the opportunity to leverage traditional visualization techniques from the rich history of printing. The characteristics of our display led us to use 2 techniques in particular: *small multiples,* in which juxtaposition of graphical depictions helps the user compare alternative possibilities, "revealing all at once a scope of alternatives, a range of options" [27]. and *micro/macro reading,* which uses the information capacity of resolution to simultaneously provide overview and de-
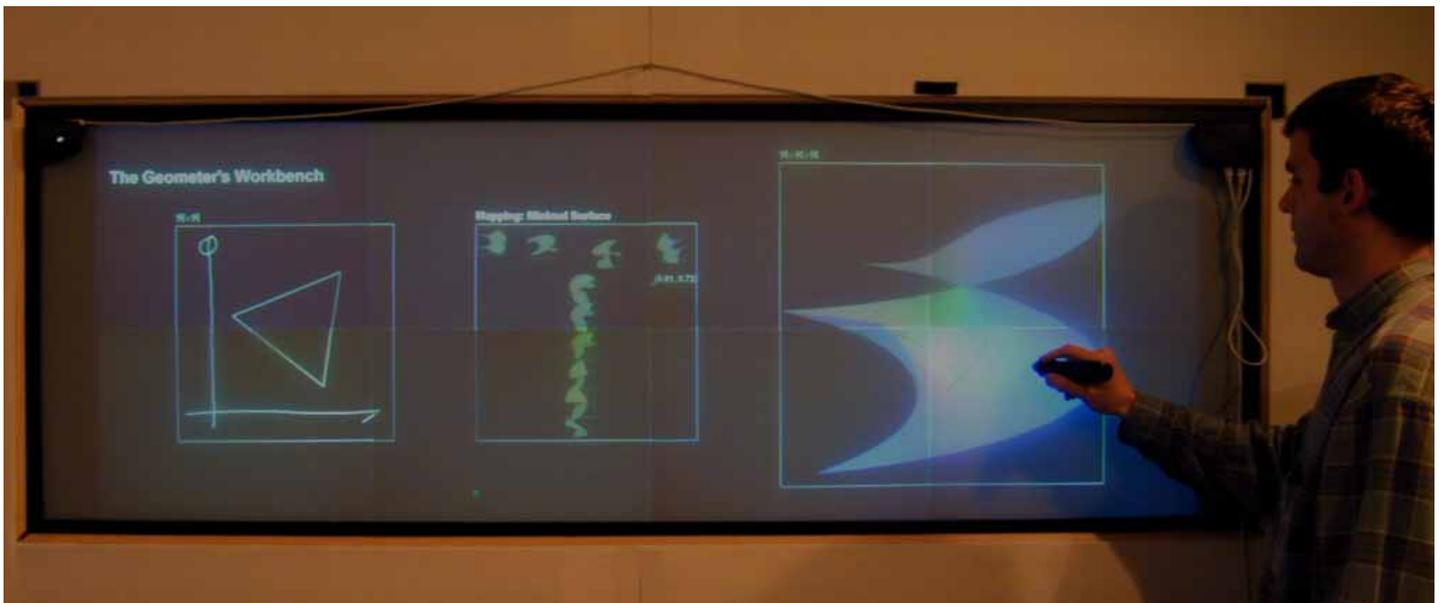
Figure 1: An overview of the Geometer's Workbench. The screen is divided into 3 sections from left to right: The 2D patch on which the user can draw arbitrary sketches; the MultiPoint area used to select a mapping, and the 3D model which shows the result of the mapping. The pods at the top corners of the screen are receivers for the EBeam pen tracking system.

tail, letting the user switch between them visually with no window-control actions.

- **Interactivity in the face of high latency** One of the prime advantages of an active display over a physical blackboard is the ability to do interactive "steering" in a simulation space. In that situation, high latency is the rule rather than the exception, since users will typically be pushing the envelope of the simulation capabilities. In our case we expect a latency anywhere between 1 and 10 seconds due to the calculations done by Mathematica. This means that simple control techniques such as sliders are not effective. We developed MultiPoint, a new widget to explore parametric spaces, which leverages the resolution and size of the display to cope effectively with latency.

- **Infrastructure** Even though the user perceives the system as one device, the system is composed of 10 computers sharing rendering, computation, and input processing tasks. This departure from an "all in one box" paradigm made it necessary for our group to develop new infrastructure for integration. That work is described elsewhere [4].

RELATED WORK

Electronic whiteboards such as Liveboard [9], are modeled closely upon analog whiteboards' passive capture of strokes. They typically are characterized by having low resolution display and limited graphic performance. Tivoli and its successors [22, 20, 21] focus on the entry of handwritten text in structures such as lists and outlines, which are typically employed in meetings. FlatLand [8] is a more evolved Liveboard application that supports small everyday tasks in a casual setting, and also supports the input and organization of written text.

More recently, several groups have created large, high performance displays. These projects have focused primarily on rendering issues. The DataWall [1] uses high end graphics machines, while projects at Princeton, UNC, and Stanford [28, 24, 14] use commodity hardware. The Stanford project focuses on delivering a high resolution (60dpi) display at which several users can stand and interact.

Symbolic algebra has a long history. Examples include MAC-SYMA [3], ScratchPad/AXIOM [17], Maple [19]and Mathematica [29]. These were designed principally for the manipulation of algebraic structure presented in linear textual form. Later systems such as MILO, MathCAD, Theorist and GraphingCalculator allow the user to drag expressions around to trigger computation, and have allowed the interactive manipulation of two-dimensional textual structures.

One of the earliest examples of graphical interaction in a geometrical content is SKETCHPAD [26], in which a user manipulated a light-pen to move line geometry. Other examples include VPS [13], Geomview [23] and Oorange [12]. All of these systems, however, represent geometry as 3D graphical structures rather than as formal structures such as those that can be defined in a symbolic algebra system. The Geometer's Sketchpad [16] supports the graphical manipulation of geometrical, as opposed to graphical, structure, but in their case, the geometry is limited to classical Euclidean geometry.

A closely related project is the *Smart Board* project for direct mathematical manipulation [2] (note that this is differ-
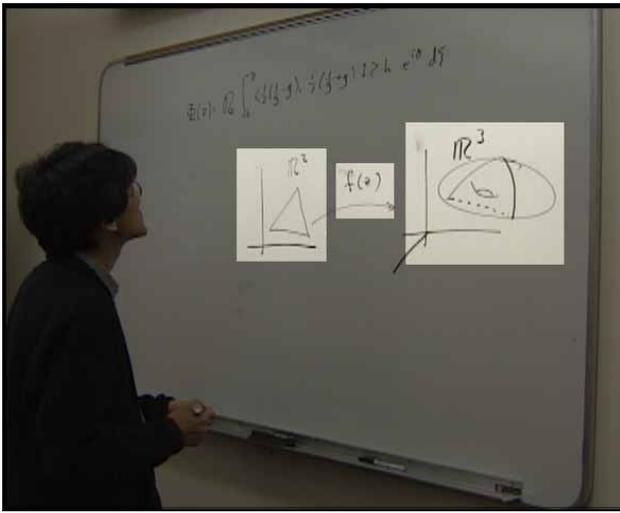
Figure 2: An example of whiteboard interaction showing a common graphical idiom used by mathematicians. From left to right the highlighted regions are: a 2D patch to be mapped (the domain), a mapping definition, and the result of the mapping.

ent from the SmartBoard commercial electronic white board product). The Geometer's Workbench is distinguished in that it makes use of a powerful, existing symbolic engine – Mathematica – and at the same time aims to support *informal* sketching and casual board work, rather than the generation or verification of proofs. This last aspect of the Geometer's Workbench also distinguishes it from theorem provers such as Analytica [6].

THE GEOMETER'S WORKBENCH

The Geometer's Workbench bridges the gap between two modes of mathematical exploration: casual whiteboard interaction during which the user mixes sketches, diagrams, formulas, and fragments but has no computational power at her disposal; and the use of an analytical engine such as Mathematica, which provides more precise results but often forces the user to use constraining syntax. The Geometer's Workbench can be thought of as a new kind of front-end to the Mathematica kernel using the Interactive Mural. For our experiment we have put aside the formula recognition problem and limited the scope of our investigation to geometrical mappings between differentiable manifolds. Rather than entering a symbolic formula, the user chooses a mapping function by picking a parametric family from a library including mappings such as sphere, saddle, and minimal surface and specifying the values of 2 parameters. She then can explore how arbitrary sketches drawn on the board will be mapped by the function onto the specified 3D manifold.

We wanted to give the same casual feel to our interface as a blackboard. Given the large area of the screen and its high resolution, contextual information can be displayed along with the current focus, rather than having to open and close (or cover over) multiple windows. In the Geometer's Work-

bench, the user does not manage windows, but sees a unified surface with three areas that each have a distinct presentation and interaction behavior.

The layout (figure 1) was inspired by a graphical idiom often used by mathematicians (figure 2). At the left of the screen is a representation of the 2D patch to be mapped (the mapping domain) (figure 3 left). In the center, we provide a graphical way to specify the mapping of interest using the MultiPoint area (figure 3 center). At the right the user can perform 3D manipulations on a rendering of the result (range) of the mapping onto the surface (figure 3 right)

LEFT: 2D PATCH The 2D patch represents the mapping domain. The user can draw freehand or select among a menu of simple forms such as circles, equilateral triangles, and rectangles, which can be dynamically adjusted in size and orientation. In our experience, users prefer to create the shape they need by sketching, rather than using predefined shapes.

CENTER: MULTIPOINT Using the middle segment, the user can select the mapping, using graphical specification rather than equation parsing. After picking one of the parametric families using a menu, the user can explore the space of parameterization using the rectangular MultiPoint window as an explicit representation of the underlying 2D parametric space. Clicking on the point (s, t) on the screen will show the resulting mapping for this pair of parameter values. Figure 3 center shows an example in which the horizontal axis (s) represents a variation in twist while the vertical axis (t) represent the amount of the surface shown. Instead of going point by point, the user can draw a stroke and the system will sample it appropriately, leaving behind a trail of non-overlapping sampling images. When the user is satisfied with the exploration, she can specify any point in the parameter space as the value to be used for mapping by using the menu.

RIGHT: 3D MODEL On the right is a shaded 3-Dimensional model of the parametric surface for the range of the mapping. As strokes are drawn on the 2D domain patch, their image under the mapping appears on this surface. The pointer can be used within this part of the board to interactively rotate the model in 3 dimensions, in order to see all parts of the curve and get a better sense of how it maps onto the surface.

Modifications made in any of the three sections are automatically reflected in the in the other two, making it easy for the user to compare different alternatives.

INTERFACE DESIGN

Out goal was to facilitate a fluid interaction experience through an interface that was appropriate to the specific affordances of the Interactive Mural. This had several consequences for the design.
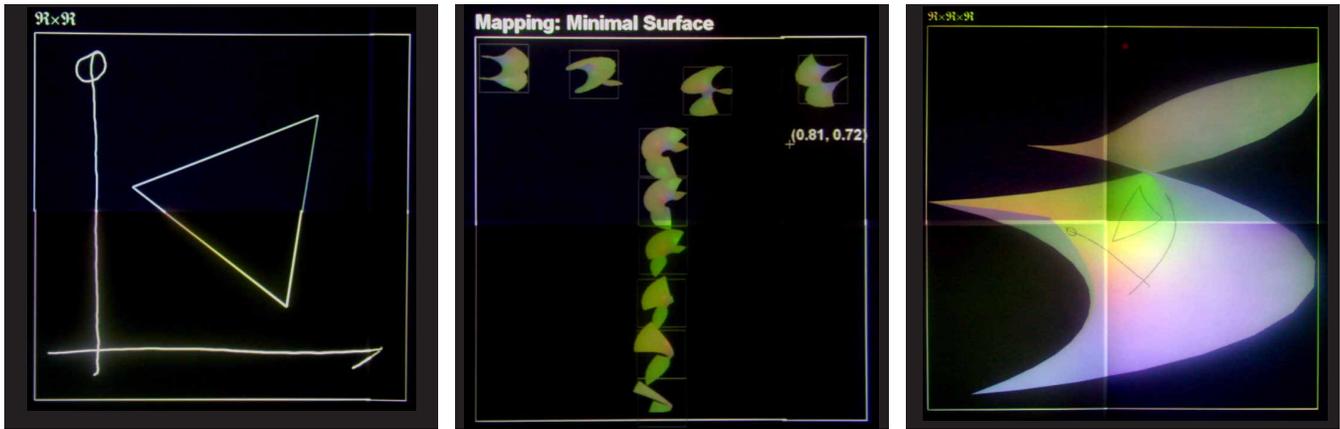
Figure 3: Close-up of the different areas on the board. **Left**: the 2D patch used to draw sketches. Using the menu the user can also draw simple geometrical shapes like the equilateral triangle shown here. **Center**: The MultiPoint area is an explicit representation of the underlying parameter space. The parametric family in this example is Minimal Surface. Pointing in the area at (s, t) will show the resulting mapping for this pair of parameter values. In this case, s represents the twist of the surface and t represents how much of the surface is used. The value selected for the current mapping is shown by a cross hair with the corresponding value attached here $(0.81, 0.71)$. Because samples stay visible. the user can easily compare different areas of the parametric space even in presence of high latency. **Right**: The 3D Model area shows the result of the current mapping. The 3D object can be manipulated using the pen as a virtual trackball.

BLACKBOARD LIKE INTERFACE

- **Gesture/stroke oriented interaction** Given that interaction was with a pen-like device, with no keyboard or indirect pointing, all actions are done through strokes on the board, interpreted either as geometry (as in drawing strokes to see how they are mapped) or as gestures indicating commands (as in sampling the parameter space).
- **Locality of interaction** All actions are done at the point of the object of interest, rather than on menus or tool bars located at edges of the screen. All menu selection uses the FlowMenu [11], a new kind of popup menu activated and accessible at the locus of attention. There are no special command modes, with the visible structure on the board always providing information about the context for action. This is particularly relevant in the case of multiple simultaneous users at a large display.
- **Minimal decoration**. The visual space is used to display task- relevant information rather than for GUI widgets or controls. The large size enables us to designate specific areas of the board for specific purposes and thereby avoid window management and its accompanying decoration (borders, scroll bars, title bars, etc.) With the Flow-Menu, artifacts for control only appear when needed. The use of the visual 2D space for parameter entry avoids the need for dialog boxes and text entry fields.

HIGH RESOLUTION AFFORDANCES

One of the exciting characteristics of devices such as the Intractive Mural is the number of pixels they provide. As mentioned earlier, our board provides as many pixels as a page printed at 300dpi. We are exploring how this resource can be used to improve the interface. We can draw on some techniques that have evolved for printed graphic design [27]:

- **Macro/Micro Reading** Macro/Micro layout simultaneously provides a macro reading at the overview level that lets the user see the larger picture, and at the micro level where the user can dig in for precise information. Our layout for the Workbench only scratches the surface of how this technique can be used to alleviate the use of multiple windows, and to some extent zooming.
- **Small Multiple** In a small multiple layout, several options are presented to the user simultaneously in the visual field to help identify differences and similarities. This technique is extremely efficient in printing but has had somewhat limited use in the computer world because it requires both space and resolution. We used this technique is the central panel, letting the user build her own small multiple display by clicking directly in the parameter space. Again, the use of visualization techniques used in synergy with the affordances of our display allowed us to keep the use of extra widgets to a bare minimum.

DESIGNING FOR LATENCY

The integrated use of a relatively slow computation engine required special attention to latency issues. In an ideal system, everything would respond fully interactively. For example, a change to the parameters to the mapping function would be reflected instantaneously in the 3-D visualization. However, although some aspects can be immediate (such as showing the image of the domain sketch and rotating the 3D model), this cannot be done for everything, due the capabilities of the mathematical computation engine and the rendering pipeline and network latencies. Mathematica can take several seconds to compute a new surface when parameters are changed. We do not see this as a temporary limitation that will be solved by faster hardware, since people will always
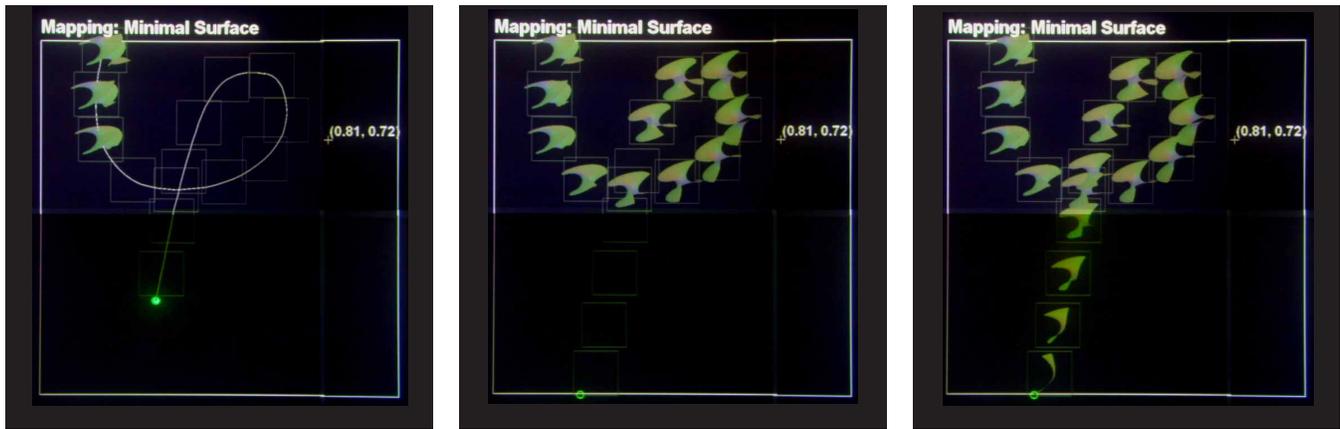
Figure 4: The Geometer's Workbench interface is designed to compensate for the latency of mathematical computations, by queuing calls to Mathematica and displaying results as they become available rather than blocking. Here we show the evolution of the display as the user draws on the Multipoint area. Notice that the pen trail is resampled by the system and that the trail of samples is updated progressively as the results of mathematical computations become available.

want to work interactively with visualizations whose computational demands are at the current edge of the computation-speed envelope, wherever that may be.

In the Geometer's Workbench we have decoupled interaction and computation. Whenever the user interacts with the system, it provides simple immediate feedback before forwarding requests to the computational engine. Requests are queued in a 2 priority queue, with high priority for requests that tend to reestablish the coherency of the screen, and low priority for those that establish the exploration context. Results are displayed as they become available.

Our second approach to latency is to use space instead of time as an exploration tool. The MultiPoint component of our system is used for exploring a 2D space of parameters so that the user can pick a mapping. Given the latency it would have been almost impossible to perform this exploration with a set of sliders each offering a temporal sampling of a one dimension parameter as the user moves the slider setting. High latency destroys the motor-perceptual coherence of this activity since the parameter changes are in jumps and are delayed from the user's action that specified them. With the MultiPoint, the parameter space is spread out spatially (in this case a 2D square) and the user select points in that space where the values are shown as small thumbnails. Since time is no longer part of the immediate linkage, latency is not as much of a problem. Moreover the different samples can be seen side by side for comparison.

IMPLEMENTATION

The implementation of this system was complicated by the departure from the "all-in-one-box" paradigm. The main program, graphic subsystem, and input server all reside on different computers connected though a network.

DISPLAY

The Workbench was implemented on the Stanford Interactive Mural [14] a 6'x2', 4066x1536 pixel back-projected display screen using a 4x2 array of XGA projectors to produce a seamless display area. Our system was updated from the version described in [14] to the following configuration: Each projector is driven by a PC with an Nvidia Quadro card and runs an OpenGL server to render the graphics on its segment of the board. The client application runs on a quad processor PC, (along with one input server and Mathematica). All of the PCs are connected via a heterogeneous network consisting of 100BaseT Ethernet and Myrinet, a fast, low latency network. The mural provides an OpenGL library allowing the applications programmer to treat the 8-projector array as a single screen on which one can draw using OpenGL commands.

INTERACTION DEVICES

We use the EFI EBeam to track the user's pen on the board. The EBeam was designed to be used as a holder for a standard whiteboard pen that leaves ink on a whiteboard surface. The computer tracking produces a virtual copy of what was written on the board. For that application, precise registration is not critical since the writer sees the actual ink and there is no hand-eye feedback that depends on the tracked location of the pen tip.

We adapted the EBeam for completely virtual interaction by improving the calibration and using a dried out felt-tipped pen to deliver a suitable physical feel against the board surface. The manufacturer's API provides us with an (x, y) position and as the ID of the pen, sampled at 60H. The system can identify up to 5 different pens, but currently only one pen can be used at a given time. Because the absorption characteristics of our screen differ from those of a whiteboard, the transmission device needed to be modified to increase the signal strength. Even so, the noise to signal ratio is still high,

especially at the border and for slow movements. We are considering using a screen on a glass substrate that should alleviate this problem.

Finally, since the menu system employs a command button, we use the button of a wireless mouse as a proxy. We will integrate the button into the physical pen device in the next version of our system.

MATHEMATICA INTERFACE
Communication with Mathematica is handled by the standard MathLink protocol [29], which is relatively low bandwidth. To avoid blocking behavior, requests are queued and processed one by one by the kernel. Requests to update the 3D model window always handled first. Mathematica's [-Graphic3D-] descriptions are transformed to OpenGL calls using the MathView3D library [18]. We modified this library extensively so that it can accommodate multiple pending contexts and have better network performance.

CONCLUSION
In the future, computer use will continue to move farther away from the conventional desktop or laptop configuration. The "one-size-fits-all" philosophy of the current desktop metaphor will not be adequate, and applications designers will increasingly be called upon to consider new devices and mixtures of devices. Improvements such as large high resolution displays will lead to rethinking the conventional wisdom of human-computer interaction.

This project illustrates some of the benefits and difficulties of taking a holistic approach to the design of applications for new device constellations. We set out to retain the blackboard experience for mathematicians, while adding the power to visualize and control complex geometrical transformations. In the process of designing the application, we needed to explore several potential input devices, to build software infrastructure to support the integration, and to design new interaction affordances suited to the task and to the devices. No one piece of this could be separated from the larger context. For example, the large high-resolution screen made it possible to trade off space for latency for the "computational steering" in a parameter space.

In our long-term plan for the Workbench, the user will also be able to specify both mapping functions and parameters by writing ordinary mathematical notation in the center area of the board, and having it parsed by a formula parser such as [25]. The current system provides a library of mapping functions from which the user can select, which is easier to use, but does not have the open-endedness that will be required in a full application. We are confident that in a relatively short time, good mathematical handwritten formula recognizers will become available and can be incorporated into the system. Another alternative we plan to explore is extending the FlowMenu [11] to provide mathematical formula commands that can be integrated with text entry and selection.

One of our primary goals in this project was to further the development of tools that can be used broadly in interaction design for large- screen high-resolution devices. Our experience has indicated that they can be used to advantage to provide interactions that feel natural to users.

REFERENCES
1. AFRL/IF. http://www.rl.af.mil/tech/programs/adii/adii_dw.html.

2. et al. Al Barr, James Arvo. The smart board project: Human-computer interaction for direct manipulation of formal systems. http://www.cs.caltech.edu/ arvo/projects/SmartBoard.

3. Richard Bogen and the Computer-Aided Mathematics Group. *Macsyma Reference Manual*, 1986.

4. Pat Hanrahan Brad Johanson, Armando Fox and Terry Winograd. The event heap: An enabling infrastructure for interactive workspaces. 2000. Submitted to UIST.

5. Xing Chen and James Davis. A scalable multi-user laser input system. 2000. Submitted to UIST.

6. E. Clarke and X. Zhao. Analytica - a theorem-prover in Mathematica. In *Automated Deduction CADE-11*, 1992.

7. M. Czernuszenko, D. Pape, D. Sandin, T. DeFanti, G. L. Dawe, and M. D. Brown. The ImmersaDesk and Infinity Wall: Projection-based virtual reality displays. *Computer Graphics*, 31(2):46–9, 1997.

8. W. Keith Edwards Elizabeth D. Mynatt, Takeo Igarashi and Anthony LaMarca. Flatland: new dimensions in office whiteboards. In *Proceedings of the CHI 99 Conference on Human Factors in Computing Systems: The CHI is the Limit*, pages 346 – 353, May 15 – 20 1999. Pittsburgh, PA.

9. S. Elrod, R. Bruce, R. Gold, D. Goldberg, F. Halasz, W. Janssen, D. Lee, K. McCall, E. Pedersen, K. Pier, J. Tang, and B. Welch. Liveboard: a large interactive display supporting group meetings, presentations and remote collaboration. In P. Bauersfeld, J. Bennett, and G. Lynch, editors, *CHI '92 Conference Proceedings. ACM Conference on Human Factors in Computing Systems. Striking a Balance*, pages 599–607. ACM Press, New York, NY, USA, 1992.

10. Electronics for Imaging. http://www.e-beam.com/.

11. François Guimbretière and Terry Winograd. Flowmenu: Continuous fluent input of selection, command, text, and parameter entry. 2000. Submitted to UIST.

12. Charles Gunn, A. Ortmann, U. Pinkall, K. Polthier, and U. Schwarz. Oorange: A virtual laboratory for experimental mathematics. Technical report, 1997.

13. Jim Hoffman. VPS, a visual programming system with visualization and rendering applications. Technical report, Mathematical Sciences Research Institute, 1996.

14. G. Humphreys and P. Hanrahan. A distributed graphics system far large tiled displays. In D. Ebert, M. Gross, and B. Hamann, editors, *Proceedings Visualization '99*, pages 215–227. IEEE, Piscataway, NJ, USA, 1999.

15. Virtual Ink. http://www.mimio.com/.

16. R. Nicholas Jackiw and William F. Finzer. The geometer's sketchpad: Programming by geometry. In *Watch What I Do: Programming by Demonstration*, pages 293–308. Cambridge, MA: The MIT Press, 1993.

17. Richard D. Jenks. The new SCRATCHPAD language and system for computer algebra. In *Proceedings of the 1984 MACSYMA Users' Conference: Schenectady, New York, July 23–25, 1984*, pages 409–416, 1984.

18. Jens-Peer Kuska. Mathview3d. http://www.mathsource.com-/Content22/Enhancements/Interfacing/Graphics/0209-056.

19. M. Monagan, K. Geddes, K. Heal, G. Labahn, and S. Vorkoetter. *Maple V Programming Guide for Release 5*, 1997.

20. T. P. Moran, P. Chiu, W. van Melle, and G. Kurtenbach. Implicit structures for pen-based systems within a freeform interaction paradigm. In I. R. Katz, R. Mack, L. Marks, M. B. Rosson, and J. Nielsen, editors, *Human Factors in Computing Systems. CHI'95 Conference Proceedings*, pages 487–94. ACM, New York, NY, USA, 1995.

21. T. P. Moran, P. T. Chiu, and W. van Melle. Pen-based interaction techniques for organizing material on an electronic whiteboard. In *Proceedings of the ACM Symposium on User Interface Software and Technology. 10th Annual Symposium. UIST '97*, pages 45–54. ACM, New York, NY, USA, 1997.

22. E. R. Pederson, K. McCall, T. P. Moran, and F. G. Halas. Tivoli: an electronic whiteboard for informal workgroup meetings. In *Human Factors in Computing Systems. INTER-CHI '93*, pages 391–8. IOS Press, Amsterdam, Netherlands, 1993.

23. Mark Phillips, Silvio Levy, and Tamara Munzner. Geomview: An interactive geometry viewer. *Notices of the American Mathematical Society*, 40(8):985–988, October 1993. Computers and Mathematics Column.

24. R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. In *Computer Graphics (SIGGRAPH 98 Proceedings)*, 1998.

25. Steve Smithies, Kevin Novins, and James Arvo. A handwriting-based equation editor. *Graphics Interface '99*, pages 84–91, June 1999.

26. Ivan E. Sutherland. *SKETCHPAD: A Man-Machine Graphical Communication System*. PhD thesis, MIT, 1963.

27. Edward Tufte. *Envisioning Information*. Graphics Press, 1991.

28. Princeton University. Immersive interactive system. http://www.cs.princeton.edu/omnimedia/index.html.

29. Stephen Wolfram. *The Mathematica Book*. Cambridge University Press, 3rd edition, 1996.