Using Large Shared Displays to Create a Collaborative Classroom Learning Environment

Patrick E. Mantey and Jeremy Richards
Jack Baskin School of Engineering,
University of California, Santa Cruz
1156 High St., Santa Cruz, CA 95064
{mantey, richards}@soe.ucsc.edu

Abstract

Our goal in this work is to create a more collaborative learning environment in the classroom, by designing and implementing a system infrastructure to support easy use and management of a large shared (tiled) display. Through this shared display space, instructor and student interaction is facilitated. This large display space provides the primary means of presentation of lecture material, allowing the lecturer to keep multiple screens of material in view of the students for reference and reuse. Additionally, students can download any portion of the tiled display they find relevant, enabling them to go back at any time to previous screens of the presentation. Students can also use this display to present their work, ideas, or questions to the class. The instructor allocates a portion (or all) of the display to students, enabling student presentation and the sharing of student work, or their responses to in-class exercises, in support of interactive group problem solving. Via their individual laptop computers, instructors and students can thus present material and interact via this tiled display, all enabled to be interactive participants, enhancing interaction and collaboration between students and instructors.

Background

The primary motivation for this work came from the first author's experience in teaching engineering courses, particularly those in the general area of "signals and systems" and including data communications, digital signal processing, computer networks, and control systems. In all of these, simulations and in-class problem solving have been found to be very valuable in enhancing student understanding of difficult concepts, bridging the gap between theories presented in class with real world applications. Having students solve problems or running simulations (using MATLAB® or other tools) in the class, and facilitating the sharing of their results with the class, is believed to enhance their learning and improve the classroom experience. Students today usually have laptop computers and bring them to class. With our classroom system and the large shared display, students can make much more extensive use of their personal laptop computers during lectures and discussion. In our "Simularium" project [1], we have equipped a 100-seat classroom with a large scale tiled display that functions like an interactive mural. This enables computer assisted in-class note taking, redirecting student efforts more to in-class learning, and also facilitates student presentation of their work to the others in the class.

The environment we have created also enables a modern and more versatile version of the classic environment, when an instructor asks students to work out a problem on the chalkboard. It enables students to evaluate different approaches to the solution of a particular problem, and to collaborate and share their results among the other members in the class. The instructor can let individual students present their results (e.g. MATLAB® graphics or simulations) to be seen by the rest of the class. In this way the students and their work become part of the learning experience, engaging with the faculty and with each other.

Current Shared Display Configuration and Expected Use

We are in the early stages of experimentation with our large shared display ("mural") in the classroom. Our current implementation of a shared display uses multiple front projection displays. With these multiple displays we have several million pixels of display space to support presentations and interaction. This changes the classroom environment even for the conventional lecturing. The large resolution of the tiled display allows the instructor to leave a significant portion (or all) the lecture in view for re-use during the lecture. This provides a convenient venue for students and instructors to ask questions or reference previously presented material, resulting in enhanced productivity for learning new material [2]. Students can use their own laptop computers in conjunction with this shared display to give them interactive control over their own personal display of lecture material, and to support their capture of lecture material, making

note taking more efficient. Students can display and save any portion of the tiled display they find relevant, and thus review material presented earlier in the lecture, thus accommodating different speeds and styles of learning. We believe that this environment will also provide additional benefits to students with visual disabilities (or those who sit in the back of a lecture hall). They can zoom into and save any portion of the tiled display, all from the comfort of a laptop computer in front of them. We are currently working on intelligent automation of the note-taking, with students – if they wish -- getting an automatic capture when the instructor finishes a "page". Similarly, the "cue" of a page change (e.g. new paper on the document camera, or a different slide in PowerPoint®) can be used to automatically move that page to another portion of the tiled display.

Prototype Design and Implementation

In our classroom context, we are primarily concerned with reliability and ease of use with the large tiled display in a real time setting. Popular solutions to drive such tiled displays include specialized graphics pipelines and Linux based PC clusters [3]. Both of these solutions can be expensive, and also difficult to operate for a novice user, reducing their value for classroom teaching. For our implementation, we are utilizing the advancements in multi-head PC graphics cards, coupled with the simplicity and affordability of a single, Windows-based, desktop computer. Our project, at least initially, employs 4 front-projection displays arranged in a linear fashion using edge alignment, providing an overall display of over 5 million pixels. The choice of front projection displays is based on the physical room layout of the Simularium, as well as economic considerations. Future plans include experimenting with high resolution plasma and LCD displays as a replacement for the front projection displays, and for use of more displays – and thus more pixels – in the tiled display. To drive the tiled display, we are using a recent, x86 dual-processor PC running Windows XP Professional, with two high end NVIDIA dual head cards that support OpenGL and DirectX hardware-based acceleration across all displays. This provides simplicity and maintainability with a single user interface, yet does not sacrifice our ability to utilize graphics across multiple displays.

Modes of Operation

To organize this process among many users, three modes of operation are required for operation and management of the large scale interactive tiled display:

- *collaboration mode*
- *laptop presentation mode*
- *instructor mode*.

Collaboration mode, illustrated in figure 1, is the default mode when the tiled display starts. Any change in mode or occupancy in the tiled display space is reflected to each end user in the audience. ***Collaboration mode*** involves audience members, with laptops or tablet PCs, who want to share the contents of their monitor to any available portion of the tiled display. This can be used for ad-hoc group research or classroom collaboration on complex engineering problems, something that is very difficult to achieve using the traditional chalkboard setting.

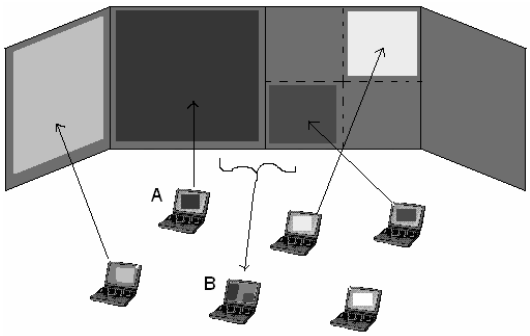

Figure 1. Collaboration Mode a) Students displaying work b) Students Capturing the Tiled Display.

The computer running the tiled display (usually under control of the instructor) manages its own allocation of tiled display space and the current mode of operation, and conveys that information to each student viewer when it changes. An important implementation issue in this environment is display space management. Our objective was to maintain a balance between an organization to support collaboration, and one that provides students a simple method to launch the contents of their laptop to the desired section of the display. The display space management functionality is based around tiling of windows, as compared to window overlay [4], using the actual tiled display boundaries as a reference. Students cannot display over or erase an existing window of another user. Each student has a small GUI of the tiled display area and information telling which tiles are in use. Students are able to launch and occupy ¼, 1, 2, 4, or N tiles in the display system. This provides for a manageable interface and general tiled display framework, not specific to our particular setup.

Laptop presentation mode involves a single person, either the instructor or another person enabled to present to the entire classroom audience. In this mode, this individual has control of the entire display space. This prevents other members of the audience from displaying the contents of their monitors on the tiled display during the presentation, as a courtesy feature. If the person giving the presentation wants the audience members to interact, he/she can easily switch to collaboration mode. Laptop presentation mode can only be cancelled using the instructor or laptop presentation mode, so presenters need to leave the presentation mode and return to collaboration mode when they are finished. Laptop presentation mode has priority over collaboration mode, while instructor mode has priority over both.

Laptop presentation mode also solves the common problem where some laptops are incompatible with the data projector used for the presentation. Using laptop presentation mode, the person giving the presentation needs to take a few minutes before the presentation to install the required *Simularium* application software (including VNC, as described below under "implementation). We plan to include this in a single installable package. With this in place, switching speakers takes virtually no time, and the previously lost time between speakers can be minimized and /or employed usefully for audience questions and discussion. This will also avoid the embarrassment situation faced by a speaker trying to get a projector to work in front of a restless audience. We expect to be able to support a wide variety of operating systems including Windows, Linux, and MacOS, for a wide variety of presenters and audience members. (This could even be extended to inexpensive personal computing devices like PDA's or cell phones, for students without laptops or tablet PCs.)

The student software used by the individuals in attendance, on their individual computers, provides a single interface that is easy to use. It displays the current mode of the system and what sections of the tiled display are available to the user. The contents of the laptop can be displayed on the large scale tiled display using a remotely launched viewer. The student's viewer has "Snag-It©"[5]- like functionality. This gives the students or members of the audience the ability to capture and digitally record any section of the tiled display at any given time. Students can be selective, archiving whatever parts of the lecture that they wish to retain and annotate. Providing the audience with a "Snag-It©"-like functionality on the contents of the tiled display server simplifies the display system software, vs. the alternative of archiving the entire display buffer as a jpeg image every few seconds and having the students download it from the machine driving the tiled display. (This is discussed in more detail later when the implementation is described.)

The student application is composed of the following functional operations moderated by the tiled display:

- Connect to tiled display
- Launch display on tile(s) X
- Kill own display
- Capture Tiled Display
- Zoom In/Out of the Tiled display
- Enter laptop presentation mode
- Leave laptop presentation mode (Return to collaboration mode)
- Allow instructor to view student's display

The motivation for ***instructor mode*** is to provide an interface to monitor, manage, and illustrate the progress of student work. Instructor mode provides the capabilities found in many commercially available networked classrooms [6] or Internet-based virtual classrooms, providing classroom management and instruction capabilities. These have been especially useful in training students on learning new software programs, but can also be used effectively in teaching programming. The instructor application allows the instructors to view the progress of the students on computer-based problems assigned to them. Unlike the commercially available management software tools, in our implementation the students control when the instructor (and the rest of the class) are able to view their screen, which we believe is important for an engineering class respecting the rights of students. The instructor runs the instructor application on a separate laptop computer, to prevent students from viewing the displays of their fellow students.

For all students that are willing to participate in the instructor mode, the contents of their computers can be displayed in a reduced size on the instructor's computer display. These thumbnail images allow the instructor to simultaneously monitor progress of multiple students. The instructor can increase the size of these smaller individual images and remotely launch them to any portion of the tiled display as desired for sharing with the larger audience.

For example, when the system is in instructor mode, the instructor could ask the class to perform a simulation on image processing using MATLAB®. The instructor is able to view the progress of the selected student applications in a convenient single interface, as shown in figure 2.

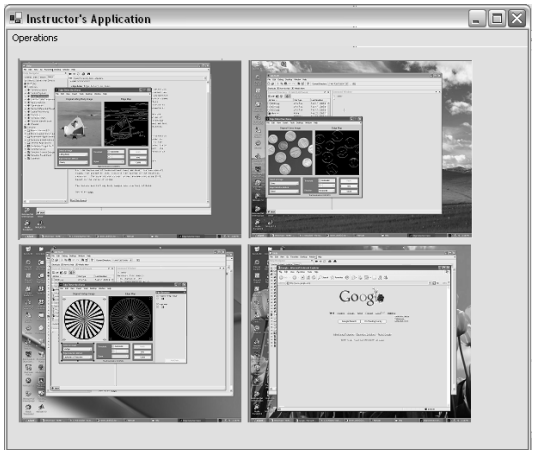

Figure 2. Illustrates the instructor's application during an in class MATLAB exercise.

The instructor can expand the size of the student's window to illustrate results of individual class work or relevant material to any section of the tiled display. In this mode, the instructor can also open a single large window on the tiled display and then sequentially scan through all the displays of the participating student's displays, using this, for example, to illustrate the different methods used for solving a question posed in class. If the instructor would like students to select their own display space management in highlighting class work, the instructor can easily switch to collaboration mode to give control back to the students.

The following illustrates the major functional operations of the instructor application.

- Enter Instructor mode
- Leave Instructor mode (Return back to collaboration mode)
- Enlarge specific display session Y to tile X
- Cycle through displays
- Kill display Y
- Kill all displays

The various modes for managing and using the tiled display space are summarized below in Table 1.

Mode	Function
Collaboration Presentation	Allows multiple students to display the content of their laptops to any available portion of the tiled display.
Laptop Presentation	Single individual has control of the entire display space. Prevents others from interfering in a presentation.
Instructor	Provides an interface to monitor, manage, and illustrate the progress of student work.

Table 1. The different modes and functions for managing the display space of a large resolution tiled display

Collaborative Taxonomy

Timothy Butler and David Coleman [7] present a general taxonomy of collaborative models based on interactivity and group size. They suggest five fundamental collaborative models of collaboration which they call: **library, solicitation, process support, community,** and **team**, as illustrated in figure 3.

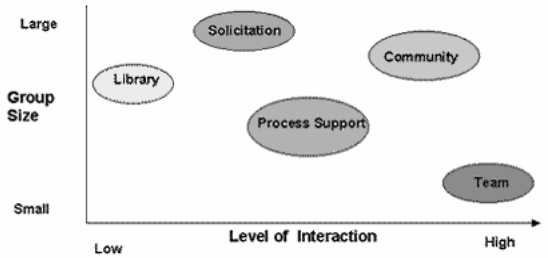

Figure 3. Collaborative Taxonomy based on Level of Interaction and Group Size

Looking at our use of the large scale interactive tiled display system in terms of their model, we have identified its use in three of the five fundamental elements presented of their taxonomy model. **Library**, **Solicitation**, and **Team** are all elements found in our interactive mural application, making the most of collaborative scenarios found in a lecture environment.

The greatest level of interaction, yet with the smallest number of participants, involves the **Team** collaborative model. This is fits our use of the tiled display in what we call the "Collaboration" mode. In this mode, the tiled display space provides a common area for members to share work, ideas or questions. Members of the group are interdependent and both read and write content to the display space. Additionally, management of the display space is organized and controlled.

The next collaborative model that matches our tiled display application involves the **Library** model. This has the lowest level of interaction, but supports a wide number of users. This is facilitated by the large resolution of the display space and the ability for lecture notes to remain in view for the duration of the entire lecture as a result of this large resolution. Students can capture the contents of the display space to create an additional repository or library of material they find relevant. As described by Butler and Coleman, the library model involves little to no feedback with the content creator. However, in our design students have the capability to save any portion in the lecture and reference that when asking a question of the lecturer, thus providing a much greater level of interaction than presented in this taxonomy. Furthermore, the indexing of lecture notes using their timestamp (and thus relative position in the lecture sequence) provides a convenient retrieval mechanism for accessing notes during such questions. Also, the number of users supported in the library model is not limited by the resolution of the display space, as in the Team model.

The last model, **Solicitation**, involves requests from a small set of requestors. The Solicitation model is represented when an instructor asks a student to present the results of their work to the class (our

"instructor" mode). Their Solicitation model is closely coupled to their Team model, and presentation of student work that will help to engage other members in the audience in questions or discussion fits into their Team model.

System Architecture and Software Implementation

In our implementation, the movement of information between the tiled display and the student's individual computer is achieved using VNC (Virtual Network Computing) [8]. Each member in the audience runs a VNC server on their individual computer. Based upon the available screen space, the end user will remotely launch a VNC viewer on the allocated section of the tiled display, which uses the address of the student's corresponding VNC server. .

VNC uses a protocol known as RFB (Remote Frame Buffer) which provides a software abstraction of a computer's frame buffer over a network connection. The RFB protocol is independent of the underlying operating system, desktop window manager, or application, since it works at the frame buffer level. VNC uses a TCP/IP connection, supports up to a 32bit color display depth, and utilizes a client pull architecture with lazy updates [9]. One of our primary motivations for using VNC is its open source availability and widespread use across multiple platforms. In our implementation, the display system on each student's laptop utilizes a VNC server, and provides an interface to remotely open and manage a VNC viewer on the tiled display based on the request of the user. In addition, the machine driving the tiled display runs its own VNC server and the students can use "Snag-It©" like capabilities from their own viewer program to record any part of the tiled display for future reference.

VNC consists of a "thick" server and "thin" client or viewer, in that the client program is relatively simple in its functionality/requirements and it is stateless. The display side of the protocol is based around a single graphics primitive: "put a rectangle of pixel data at a given x,y position". A sequence of these rectangles makes a frame buffer update and an update represents a change from one valid frame buffer state to another, so in some ways is similar to a frame of video.

In a summary, the shared display system utilizes a VNC server on each student's laptop, and provides an interface to remotely open and manage a VNC viewer on the tiled display based on the request of the user. In addition, the machine driving the tiled display runs its own VNC server, and the students are able to use "Snag-It©" like capabilities from their own viewer program to record any part of the tiled display for future reference.

Conclusions

Our work with a large shared (tiled) display in the classroom, while still in its early stages, is very encouraging. We believe it provides classroom enhancements (at modest cost) that make the classroom a much more effective collaborative learning environment, especially for large engineering classes.

References

[1] P. Mantey and J. Richards, The "Simularium" – A Large Scale Interactive Tiled Display System for Collaborative Learning, Proceedings of IASTED International Conference Education and Technology, July 4-6 2005, Calgary, Alberta, Canada

[2] M. Czerwinski, G. Smith, T. Regan, B. Meyers, G. Robertson, G. Starkweather, Toward characterizing the productivity benefits of very large displays, *Proc. Interact 2003*.

[3] M. Hereld, I. Judson, and R. Stevens, Introduction to Building Projection-based Tiled Display Systems, *IEEE Computer Graphics and Applications*, Vol. 20, no. 4, Spring/Summer 2000, pages 22-28

[4] D. Hutchings, J. Stasko, Revisiting Display Space Management: Understanding Current Practice to Inform Next-generation Design. *Proc. Graphics Interface* (2004), pages 127-134

[5] SnagIt: A popular Windows Screen capture Program, http://www.techsmith.com/

[6] Net Support School: http://www.netsupport-inc.com/nss/netsupport_school_overview.htm

[7] T. Butler and D. Coleman, Models of Collaboration**,** Collaborative Strategies LLC. 2003

[8] T. Richardson, Q. Stafford-Fraser, K. Wood, A. Hopper, Virtual network computing, *IEEE Internet Computing*, vol.2, no.1, Jan.-Feb. 1998, pp.33-8

[9] A. Lai, J. Nieh,Limits of wide-area thin-client computing, *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, Marina Del Rey, California, 2002, 228-239