T3: A Toolkit for High-Resolution Tabletop Interfaces

Philip Tuddenham and Peter Robinson
University of Cambridge Computer Laboratory
15 JJ Thomson Avenue, Cambridge CB3 0FD, UK
<Firstname>.<Lastname>@cl.cam.ac.uk

ABSTRACT

Most multi-person tabletop interfaces use a single projector to create a table-sized display. The resulting low display resolution prohibits many interesting applications, such as collaborative reviewing of text documents. T3 is a novel Java toolkit that uses multiple projectors and blending to create a seamless high-resolution tabletop interface. It allows objects to be positioned and oriented in a single large coordinate space and handles multi-person input event streams. T3 allows researchers to create prototype multi-person tabletop applications quickly, and enables experimentation with a new range of tabletop applications that previously have been very difficult to create.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces – *Computer-supported cooperative work*

General Terms

Design, Experimentation, Human Factors

Keywords

High-resolution multi-projector tabletop toolkit

1. INTRODUCTION

Tabletop interfaces have been the subject of considerable research in recent years, and the CSCW literature abounds with a diverse range of tabletop applications, from planning room layouts to organizing digital photos.

However, almost all these projects use a single projector to create a table-sized display. This results in a low display resolution that severely limits the range of applications. For example, this low resolution prohibits small text, and so there has been little investigation of collaborative writing and reviewing of text documents on tabletop interfaces, despite the fact that these are daily tasks for which many knowledge workers currently use their physical meeting room tables [2]. Similarly, there has been little investigation of collaborative spreadsheets or programming interfaces on tabletop interfaces.

How, then, should we create a large high-resolution tabletop interface? A collection of commodity LCD monitors can be tiled to create such a display [1], but this approach results in seams between panels, and precludes the use of a touch-sensing surface or a digital stylus. High-resolution projectors are very expensive and unsuitable for tabletop interfaces. Our approach is to use several projectors and a single commodity desktop PC to create a high resolution display on a small table.

CSCW '06, November 4-8, 2006, Banff, Alberta, Canada.

In order to prototype CSCW applications rapidly using a multi-projector display, we need a software infrastructure presenting a useful API. This handles the necessary coordinate transforms, and multi-user input event streams, just as the widely-used DiamondSpin [3] open source toolkit does for single-projector tabletop displays.

However, the DiamondSpin toolkit cannot easily be extended to perform the warping and blending necessary to drive a multi-projector display. Several display wall toolkits [e.g. 4] are available to drive multi-projector displays but are designed for high-performance distributed rendering tasks and are not suitable for supporting CSCW research prototyping.

In this poster we outline the challenges involved in creating a high-resolution tabletop interface and present the TableTop Toolkit (T3). T3 is a novel Java toolkit that enables CSCW researchers to build prototype applications for a high-resolution tabletop interface, in the same way that DiamondSpin allows them to build prototype applications for low-resolution tabletop interfaces. T3 is freely licensed to academic researchers and allows them to investigate a new range of applications for multi-person tabletop interfaces.

2. ARCHITECTURE

We use six projectors and a single PC with three dual-head graphics cards to create a 4.7 Megapixel display over an area of 0.5m^2. All these components are available off-the-shelf and none are high specification. The six projectors generate little noise and heat and do not distract from collaboration. We use digital styluses for multi-user input, and the system could easily be extended to incorporate a multi-touch surface.

Our requirements for the software toolkit are based on the requirements of the DiamondSpin project [3] and the experiences of other recent tabletop projects, as well as our own needs in investigating new applications. T3 presents an API based around a single seamless large display, allowing the application programmer to arbitrarily position and orientate objects within a single large coordinate space. T3 then creates the correct images to send to each of the six projectors in order to create the illusion of a single seamless display to the users sitting around the tabletop.

As with all multi-projector displays [e.g. 4], T3 must apply small warps and blending masks to each of these images sent to the projectors in order to compensate for projector keystone and slight misalignments in projector mounting, and to create seamless transitions between the projector outputs. We perform this warping and blending using the graphics cards, via Java OpenGL calls. We achieve a frame rate of 50fps with no noticeable latency, even when large areas of the display are updated. These warps and blends are configured in a quick semi-automatic calibration procedure.

3. USING THE TOOLKIT

T3 is written in Java. Its API is based around rectangular areas of pixels called *tiles*, which are analogous to windows on a conventional desktop computer interface. The applications programmer creates tiles and specifies how each tile is positioned, rotated and scaled on the display surface, relative to a single large coordinate space. The programmer also specifies how users can drag and rotate each tile.

Updating the contents of a tile is as simple as painting a window in a conventional computer program. The application programmer can use Java Swing components, such as text fields, radio buttons and third party components such as spreadsheets. Alternatively, the programmer can draw directly into the tile's rectangular image using the usual Java Graphics class.

```
// create a new tile of 500*500 pixels, on tabletop d
myTile = new SwingFramePortfolio(
      d, d.rootPortfolio,  500,500);

// add a JSpreadsheet and a menu bar to the tile's frame
myTile.getFrame().add( new JSpreadsheet(80,40) );
myTile.getFrame().setJMenuBar(menuBar);

// make the tile's size on the display 200mm*200mm
myTile.setTileWidthAndHeightInPORT(200.0, 200.0);

// position the tile 300mm from the top and left of the tabletop
// rotate it by 0.17 radians and set scale factor 1.00.
myTile.setPORTtoPPORT(300.0, 300.0, 0.17, 1.00);

// make the tile visible
myTile.setVisibleWhenParentVisible(true);
```

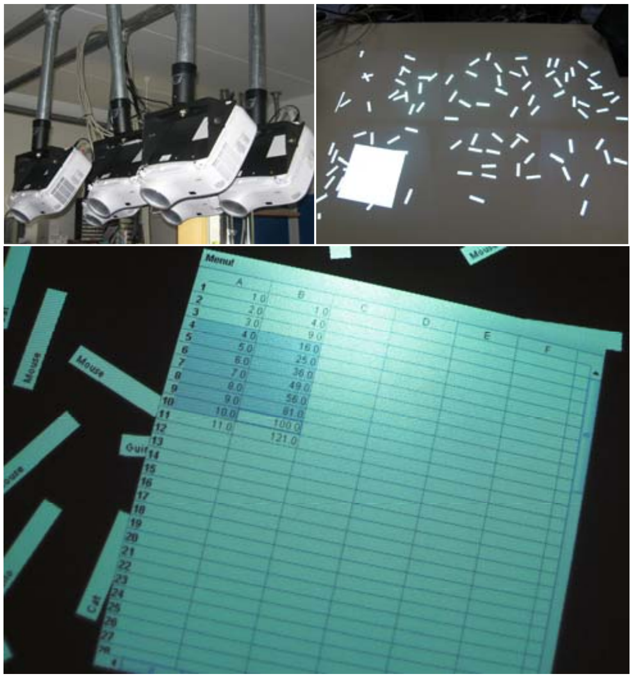

Figure 1. Tabletop interface showing rotated spreadsheet and a magnetic poetry application. Sample code (top), projector rig (mid-left), display surface (mid-right) and close-up of spreadsheet (bottom).

Based on the application programmer's tiles, T3 then creates the correct images to display on each projector, and warps and blends the resulting output to achieve a seamless display.

T3 handles the simultaneous input event streams from multiple users, dispatching each event to the appropriate tile, where it can be handled by the applications programmer's event handling routines. Events are represented as subclasses of the Java AWT event classes. T3 transforms the coordinates of each event to be local to the tile's rectangular area, undoing the effect of the positioning, rotation and scale within the display. Java Swing components respond as expected to the events without any intervention from the applications programmer. Two users of the tabletop can work simultaneously on different tiles, or on the same tile; there is no concept of an "active tile".

Figure 1 illustrates use of the toolkit to create a tile containing a standard Swing JSpreadsheet component. The sample code results in the output shown in the photos.

4. CONCLUSIONS AND FUTURE WORK

We have presented T3, a novel Java toolkit that allows CSCW researchers to create prototype applications quickly for a large high-resolution tabletop interface constructed from commodity components. This enables experimentation with a new range of tabletop applications that previously have been very difficult to prototype. We have shown that T3 is easy to use and performs well, and hope that it will be adopted by third parties as a mainstream CSCW research platform.

T3 is versatile, and we are currently using it to prototype applications in visualizing maps, large collaborative spreadsheets, collaborative web browsing, programming interfaces and collaborative reviewing of text documents. We are also adapting it to investigate scenarios in which two geographically-separated tabletops are linked via a computer network to provide a shared workspace for both co-located and remote collaboration.

5. ACKNOWLEDGEMENTS

We gratefully acknowledge the support of Thales Research and Technology, who are funding this work.

6. REFERENCES

[1] Krumbholz, C., Leigh, J., Johnson, A. et al. Lambda table: High resolution tiled display table for interacting with large visualizations paper. In *Proc. Workshop on Advanced Collaborative Environments*, 2005.

[2] Sellen, A.J., and Harper, R.H.R. *The Myth of the Paperless Office.* MIT Press, 2003.

[3] Shen, C., Vernier, F. D.. Forlines, C. and Ringel, M., DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction. In *Proc.CHI'04*, 2004.

[4] Wallace, G., Anshus, O. J., Bi, P. et al. Tools and Applications for Large-Scale Display Walls. *IEEE Computer Graphics and Applications, 25, 4* (July/August 2005), 24-33.