T3: Rapid Prototyping of High-Resolution Tabletop Applications

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

Multi-person tabletop applications that require a high display resolution, such as collaborative web-browsing and collaborative spreadsheets, are currently very difficult to create. As a consequence, there has been little investigation of important tabletop applications, despite promising early results. In this position paper we present T3, a software toolkit that addresses these challenges. T3 allows researchers to rapidly create high-resolution multi-person tabletop applications for co-located collaborators. It uses multiple projectors to create a single seamless high-resolution tabletop display. Furthermore, it supports existing user interface components, including buttons, web-browsers and spreadsheets, allowing the rapid creation of complex tabletop applications. We show how we have used T3 to create five novel tabletop applications that would previously have been very difficult to build.

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 web-browsing, spreadsheets, programming interfaces or document review on tabletop interfaces, despite the fact that these are compelling and popular applications to which tabletop interfaces might well bring significant benefits.

Researchers are prevented from investigating these applications by two problems. Firstly, by far the easiest way to create a higher-resolution interface is to use multiple projectors in a tiled array, but this introduces technical challenges. For example, we have created a 4.7 Megapixel display using 6 projectors and a single off-the-shelf PC. The huge number of pixels can lead to unresponsive performance. Furthermore, it is impossible to align the projectors to the degree of precision required to create a perfect seamless display, and so these displays can suffer from small overlaps, mismatches and keystoning. To compensate, small adjustment transformations and blending masks must be applied to each frame before it is sent to the projectors. These solutions are well-understood and employed in the software toolkits [e.g. 5] used to drive multi-projector display walls, but such toolkits are not suitable for supporting CSCW research prototyping.

Secondly, any software to support the rapid creation of complex high-resolution tabletop applications such as spreadsheets or web-browsing must be able to support existing user interface components. This is essential because, for example, it is not within the scope of a CSCW project to reengineer from scratch complex applications like web-browsers or spreadsheets.

Two software frameworks are currently available to assist researchers in rapid creation of tabletop applications. Both address issues such as allowing collaborators to arbitrarily position and orient artifacts, and allowing multiple users to interact concurrently. The DiamondSpin toolkit [3] is widely used and supports existing user interface components, but cannot easily be extended to a multi-projector tiled display because of its rendering architecture. By contrast, the Buffer Framework [1] is designed to provide good performance on a multi-projector display when hundreds of digital artifacts appear on the tabletop, but it does not permit reuse of existing user interface components and so creating new complex applications is difficult. To our knowledge, it has thus far been used only to create an application for moving and grouping photos.

In this position paper, we present T3, a novel software toolkit that we have created to address these problems [4]. T3 enables CSCW researchers to rapidly build complex applications for high-resolution tabletop interfaces, in the same way that DiamondSpin allows them to build applications for low-resolution tabletop interfaces. We show how T3 has enabled students in our laboratory to create four novel applications that would otherwise be very difficult to investigate. T3 is freely licensed to academic researchers.

2. ARCHITECTURE

We use 6 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 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 and the experiences of other recent tabletop projects. T3 presents an API based around a single seamless large display, allowing the application programmer to arbitrarily position and orientate digital artifacts within a single large coordinate space. T3 then generates 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. It uses OpenGL hardware-accelerated rendering and applies small corrective transformations and blending masks as appropriate. For the applications we tested, we achieve frame rates of 10-30fps depending on the level of activity and the number of artifacts on the display.

3. USING T3

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 application 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 whether users can drag and rotate each tile using Rotate'N'Translate [2].

Existing Java Swing user interface components can be used in T3 without any modification required on the part of the applications programmer. This supports the creation of tiles that contain basic widgets such as radio buttons and text boxes, and even third-party components such as spreadsheets and word processors, enabling rapid construction of complex tabletop applications.

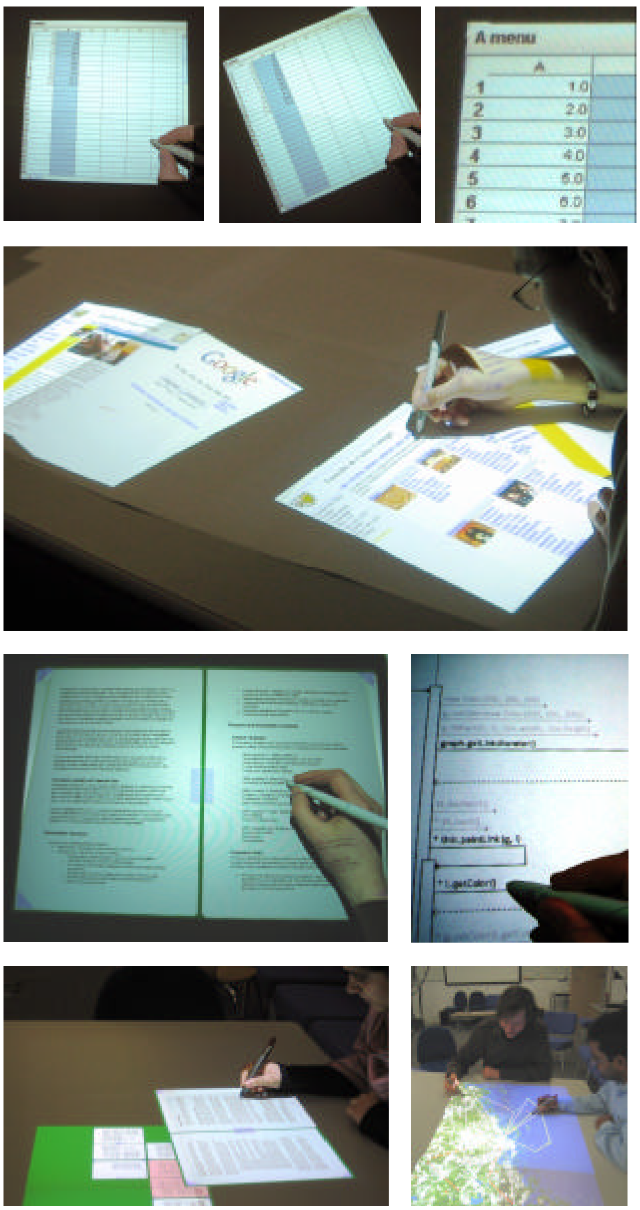

Figure 1. Tabletop applications created using T3.

T3 handles the simultaneous input event streams from multiple users, dispatching each event to the appropriate tile and transforming the coordinates of each event to be local to the tile's rectangular area. Multiple users can interact simultaneously on different tiles, or on the same tile; there is no concept of an "active tile".

4. T3 APPLICATIONS

Figure 1 shows four research projects currently being undertaken by students in our Laboratory. The projects illustrate the utility and versatility of T3, and would be very difficult to implement without the core functionality that it provides.

The collaborative tabletop spreadsheets application (top) is an example program that uses just 6 lines of Java to create a tile and fill it with a third party Swing JSpreadsheet component. The result is a working spreadsheet that appears as a legible 20cm x 20cm tile on the table and can be passed between collaborators and reoriented. It is sufficiently small that multiple spreadsheets can be viewed simultaneously on the tabletop by multiple collaborators. Columns and rows can be selected, and numbers and formulae can be entered.

Further projects investigate collaborative web-browsing (middle upper), document review meetings (middle left and bottom left), programming interfaces (middle right), and command and control interfaces (bottom right).

T3 is freely available to academic researchers, and we hope that it will be adopted as a mainstream CSCW research platform.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Isenberg, T, Miede, A. and Carpendale, S. A buffer framework for supporting responsive interaction in information visualization interfaces. In *Proc. Creating, Connecting and Collaborating through Computing*, IEEE, 2006.

[2] Kruger, R., Carpendale, S., Scott, S. D., and Tang, A. Fluid integration of rotation and translation. In *Proc. CHI'05*, 2005.

[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] Tuddenham, P. and Robinson, P. T3: A Toolkit for High-Resolution Tabletop Interfaces. In *Ext. Abstr. CSCW'06*, 2006.

[5] 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.