Multi-User, Multi-Display Interaction with a Single-User, Single-Display Geospatial Application

Clifton Forlines, Alan Esenther, Chia Shen, Daniel Wigdor, Kathy Ryall

TR2006-083 October 2006

Abstract

In this paper, we discuss our adaptation of a single-display, single-user commercial application for use in a multi-device, multi-user environment. We wrap Google Earth, a popular geospatial application, in a manner that allows for synchronized coordinated views among multiple instances running on different machines in the same co-located environment. The environment includes a touch-sensitive tabletop display, three vertical wall displays, and a TabletPC. A set of interaction techniques that allow a group to manage and exploit this collection of devices is presented.

ACM Symposium on User Interface Software and Technology

Multi-User, Multi-Display Interaction with a Single-User, Single-Display Geospatial Application

Clifton Forlines[1], *Alan Esenther*[1], *Chia Shen*[1], *Daniel Wigdor*[1,2], *Kathy Ryall*[1]

[1]Mitsubishi Electric Research Labs
Cambridge, MA 02139 USA
{ forlines, esenther, shen, ryall }@merl.com
www.merl.com

[2]Department of Computer Science
University of Toronto
dwigdor@dgp.toronto.edu
www.dgp.toronto.edu

ABSTRACT

In this paper, we discuss our adaptation of a single-display, single-user commercial application for use in a multi-device, multi-user environment. We wrap Google Earth, a popular geospatial application, in a manner that allows for synchronized coordinated views among multiple instances running on different machines in the same co-located environment. The environment includes a touch-sensitive tabletop display, three vertical wall displays, and a TabletPC. A set of interaction techniques that allow a group to manage and exploit this collection of devices is presented.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design**,** Human Factors

Keywords: Tabletop computing, multi-display, multi-user, geospatial applications, CSCW, co-located collaboration

BACKGROUND AND MOTIVATION

In recent years, geospatial applications [4,6,7] have generated a large amount of interest. These interactive atlases allow the user to navigate to any position on the globe, and augment satellite images with geo-registered information such as roads, hotel locations, and political boundaries. Using these tools, individuals plan trips, locate businesses, or simply explore new areas from a bird's-eye-view.

Teams of intelligence analysts, civil engineers, architects, and city planners often work face-to-face around a tabletop on which they view and manipulate printed geospatial information, often surrounding themselves with additional materials hung on the walls of the room. Such a team wishing to use one of the geospatial applications described above will run into many problems, which stem from the single-vertical-display, single-user assumptions that most application developers make.

In this paper, we present an adaptation of Google Earth [4] that meets the needs of small groups working in a co-located, multi-device environment, such as that described by Streitz et al. [10]. We describe several of the challenges that arise with this type of adaptation, and present a set of interaction techniques that mediate these issues. Our wrapper application coordinates multiple instances of Google Earth running on different machines in the same space. No changes to the Google Earth application code are needed. While we believe that ultimately developers will design and implement their applications with multi-device environments in mind, we feel that the adaptation of existing tools to take advantage of these spaces is a worthwhile endeavor.

RELATED WORK

Adapting single-user applications for multi-user settings has been the subject of much research. Greenberg [5] surveyed and discussed a large body of research performed with the goal of providing shared-views across distributed sites. In this work, the research focuses on ensuring that the *same* view is displayed on different *remote* machines so that separated users have a shared context for collaboration. A more recent research effort reported by Tse et al. [11] proposed an architecture that enables multi-user speech and gesture interaction for co-located collaboration on a *single* shared-display with existing applications.

Our approach in this paper is to help groups take advantage of *different* views displayed on *multiple* machines in a *co-located* space. Much of the state of the Google Earth instances is synchronized across machines; however, subtle differences are maintained and controlled by the users so that each of the machines' displays and input devices is best exploited.

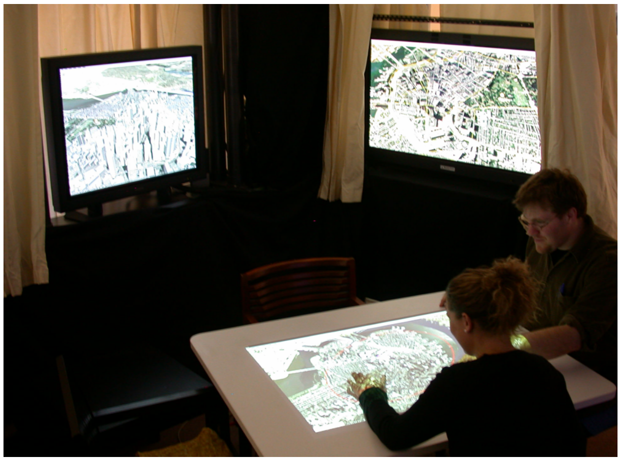

Figure 1: Two people working with our multi-device, multi-user adaptation of Google Earth [4].

UIST'06, October 15–18, 2006, Montreux, Switzerland.
Copyright 2006 ACM 1-59593-313-1/06/0010...$5.00.

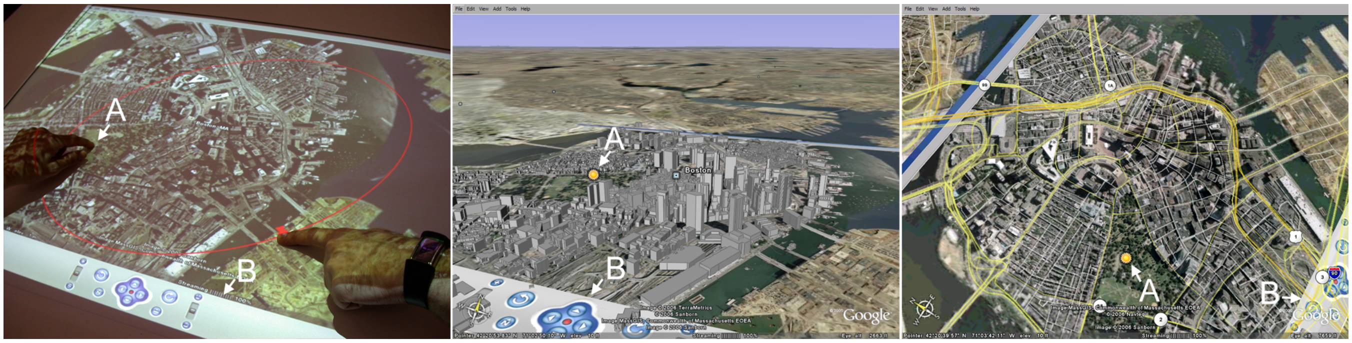

Figure 2: (note: the letters A & B are added to these images for clarity and are not part of the interface) All displays in the space focus on the same geospatial location from different views with different layers of information. The table (*left*) shows a bird's-eye-view, the first wall display (*center*) shows a ¾ view with 3D buildings overlaid, and the second wall display (*right*) shows a wider view overlaid with roads. Dragging a proxy for the camera on the table controls the view of a wall display (red camera and control ring in the left image). Telepointers (*A*) on each wall display indicates the position touched on the table by the user in the left image. The navigation and title-bar displayed on the table are replicated on other displays (*B*) to help users understand the geometric relationship among views.

A NEW INTERFACE FOR MULTI-DEVICE, MULTI-USER GEOSPATIAL EXPLORATION

Our wrapper application adheres to the premise that the underlying Google Earth (GE) application is not modified. While this is a constraint, we were able to add new interaction techniques that enable multi-device, multi-user operation to this single-display single-user application. A separate instance of our wrapper runs on each of the machines in the environment. GE downloads and renders geospatial information, including satellite photographs and registered layers of information, such as roads, hotel locations, country boundaries, etc. Additional items can be placed into the GE rendering pipeline through their KML file specification (http://earth.google.com/kml/). Our C# wrapper handles multi-user input, synchronization among instances of GE, the rendering of some interface elements, annotation and the sharing of annotations, as well as conflict mediation. We use a multi-user touch-table as our primary input device, and control the application through a combination of single-finger, mouse-like input and bimanual gestures. While there is no theoretical limit to the number of instances that can run together, we have tested our wrapper in an environment consisting of one touch-table, three wall displays, and a TabletPC.

Synchronized and Coordinated Views

The primary function of our wrapper is to synchronously coordinate the views among the multiple machines in the space. When input is performed on one machine, our wrapper polls Google Earth for the current point-of-view (POV) displayed in the application. This POV contains a latitude, longitude, altitude, tilt, and azimuth. This POV is then transformed and sent over the network to each of the remote machines, which then set the POV of the local GE instance. The phrase "synchronized and coordinated views" specifically refers to our design that the POV from the machine on which input is performed is modified before it is sent to the other machines in the space. In general, this modification never changes the latitude and longitude that is the focus of the POV; consequentially, all displays in the space are focused on the same position on the Earth.

The transformation of POVs is informed by the differences among the physical displays of the different devices in the space. A POV that is appropriate for one type of display may be inappropriate for another. For example, using a POV other than a bird's-eye-view on a horizontal tabletop results in a disorienting image for many of the group members sitting around different sides of the table. A tilted view of the scene is more appropriate for a vertical display with which everyone in the room shares the same "up" vector. Figure 2 shows the table and two walls focused on the same geospatial location, but viewing it from different POVs that are appropriate to each display's orientation.

Visually Separating Layers

Google Earth provides users access to many geospatially registered layers of information. Figure 2 illustrates how each display in our environment can be augmented with a different layer of information so that the end result is a spatially multi-plexed, temporally-synchronized multi-layered display. In a multi-user visualization application, each member of a group may find different layers of information valuable to their participation. For example, a plumber may be most interested in the location of pipes while an electrician would want to know the location of buried wires under city streets. Simultaneously displaying these multiple layers on a single-display would produce a visually cluttered and less-useful image, and switching amongst these layers sequentially might also impose high levels of cognitive load on the users. As pointed out by Roberts [9], by simultaneously displaying the data in multiple ways a user may understand the information through different perspectives, overcome possible misinterpretations and perform interactive investigative visualization through correlating the information among views. Currently, the visibility of layers on each wall display is controlled locally on that machine, but future work will include methods for controlling the visibility of layers on the wall displays from the tabletop.

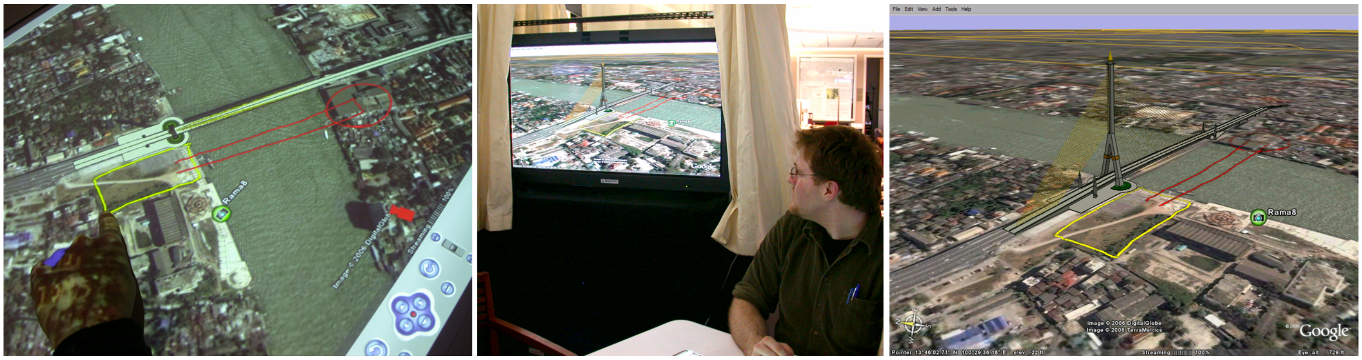

Figure 3: Multi-user annotations across different displays. (*left*) The strokes from each user around the table are drawn in different colors. (*center*) Annotations made on one display are geospatially registered so that they appear in the correct location on all machines. (*right*) Close-up of the wall display.

Controlling View Transformations from the Table

By default, our table displays a bird's-eye-view of the location of interest and each wall display presents a transformation of this POV that is based on the spatial relationship between the physical wall and table. This default POV transformation creates a non-contiguous CAVE-like collection of views [1], which feels like the most natural relationship between views, although not the only useful one.

On the table, each wall display is represented with a small proxy camera whose position indicates the wall display's rotation and tilt relative to the table's POV. Users manipulate this proxy to edit the POV transformation. By changing views in this manner on the table, users not only avoid the need to stand up and walk to a distant display to manipulate its view, but also are able to perform input on wall displays that may have no input mechanism of their own.

Dragging a proxy around the center of the table in either direction rotates the wall's POV, and dragging the proxy toward or away from the center of the table changes the tilt of the wall's POV. By constraining POV transformations to rotation and tilt, there is a natural mapping between the 2-DOF touch input on the table and two values in the POV.

While altering the POV on wall displays allows a group to view different angles of a scene, it can quickly become difficult to understand the relationship between what is displayed on the walls and the table once a group changes the initial CAVE-like arrangement. We lessen this confusion with two pieces of visual feedback to help group members understand the relationship among the displayed POVs. When camera proxies on the table are touched, an ellipse is drawn on the table indicating the tilt and orientation of the remote display (Figure 2). The table's title-bar and navigation-panel appear on the wall display as if the wall display were projecting a view of the tabletop.

Detaching & Re-synchronizing Views

Oftentimes, groups may want to break the coordination between a wall display and the table. For example, a group examining potential sites for the construction of a new bridge may want to leave one site visible on one wall while they navigate to and inspect a second site. In this way, the two sites can be examined side-by-side. Similarly, a group will often switch between working together and working separately, so being able to unlink views is valuable for these periods of independent work. Our interface provides a simple tacking gesture to lock a camera proxy in-place and break the link between the table and wall application states [3]. Subsequent navigation on other machines does not affect the POV of the tacked wall display. To re-link the wall display, a second tacking gesture refocuses the wall display on the table's point of interest. An unimplemented alternative would be to click-and-hold a tacked proxy to return all machines to the POV of the tacked wall.

Annotations

The simple act of annotation is a major activity with geospatial information. Our application implements a simple multi-user annotation tool on the table and a single-user annotation tool on machines that support input. Our tool paints into a transparent overlay and polls Google Earth for the geospatial location of the current point-of-view so that annotation strokes can be geospatially registered. This way, when the team is finished annotating, the strokes are passed over the network to other machines and displayed in the correct location regardless of the machine's current or future POVs. Specifically, we create a KML file detailing geospatial location of the bitmap file, and send both files over the network to the other machines. Other machines then open the KML file, which inserts the bitmap into the GE rendering pipeline. Figure 3 shows the sharing of geospatially registered annotations across different displays.

Working Together, Working Alone

A major benefit of a shared display is the common context that it provides the group; however, situations arise in which the use of a shared display is a liability. Google Earth, like other single-user applications, assumes that it is safe to repurposes large portions of the screen for subtasks. If the (single) user's mind is occupied on a task, what is the danger in annexing much of the display for that task? For example, when a user wishes to turn layered information on and off, they open the layer control panel, which occupies up to 1/3 of the screen. If the team is working closely together, the opening of this menu would be expected and not cause an interruption; however, when working independently, opening this large menu is a disruptive action for other members of the group.

Our interface mediates this particular disruption by moving the layers menu off of the table and onto a group member's TabletPC (Figure 4). Our wrapper application polls the visibility of each layer in the GE application, and upon detecting a change sends a message to the appropriate machine. In this manner, the visibility of all of the layers is kept in sync between the table and tablet, allowing an individual to issue these commands without annexing large portions of the tabletop. A single shared display may also have the drawback of discouraging exploration on the part of individual group members. By allowing an individual to annotate privately on their TabletPC, they may be more comfortable to try out variations on an idea in private before they are ready to share their sketches with the group.

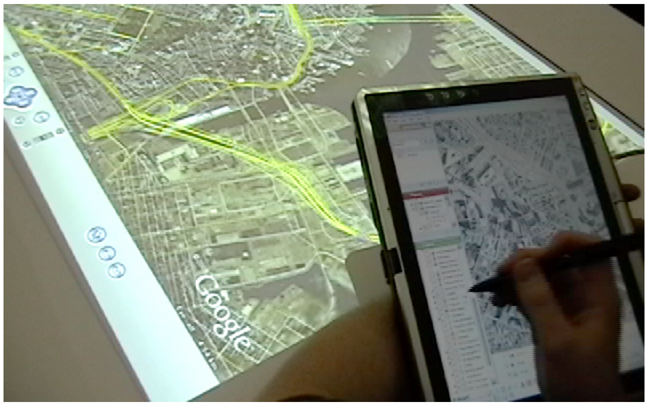

Figure 4: This user operates the layers menu on their TabletPC to view the roadways in this scene (yellow lines). The visibility of layers is synchronized between the tablet and table, the result being that this user can issue commands without covering large portions of the shared tabletop.

Touching to Navigate, Touching to Reference

Because most applications assume a single user, many commands are not well defined when performed simultaneously by multiple people. In our example application, how should the system respond when one person navigates to the left, and another to the right? By making one person the "driver", others' touches can reference items or annotate rather than change the view. "Driver" status can be passed around. First to touch is a simple means of conflict resolution for driver designation [8]. For those not driving, telepointers [2] are used to visualize touch-through from the table to the wall displays. Figure 2 (*left*) shows a user touching on the table (*at point A*) to reference an item shown on the wall (*center, right*).

CONCLUSION

Enabling multi-device multi-user interactions using widely deployed single-user commercial off-the-shelf applications creates new uses for existing tools. In this paper, we presented our adaptation of the Google Earth application. While many of the portions of this system are specific to Google Earth, others were designed with the wrapping of other applications in mind. We hope that this tool will provide an example for how other single-display, single-user applications can run within a multi-heterogeneous-display, multi-user environment without the need for major code modification, and we are currently investigating generalizing some of these solutions for use with other prototypes.

ACKNOWLEDGMENTS

This work was partially supported by the Advanced Research and Development Activity (ARDA) and the National Geospatial-intelligence Agency (NGA) under Contract Number HM1582-05-C-0028. The views, opinions, and findings contained in this report are those of the author(s) and should not be construed as an official Department of Defense position, policy, or decision, unless so designated by other official documentation.

We would like to thank Google Inc. for providing us with a Google Earth Pro license, Ravin Balakrishnan from the University of Toronto and Ed Tse from the University of Calgary for their valuable contributions, and Tim Case from Parsons Brinkerhoff Inc. for his thoughtful discussions and for introducing us to the Google Earth team.

REFERENCES

1. Cruz-Neira, C., Sandin, D., DeFanti, T., Kenyon, R., and Hart, J., The CAVE: Audio Visual Experience Automatic Virtual Environment, *Communications of the ACM*, Vol. 35, No. 6, June 1992, pp. 65-72.
2. Engelbart, D. and English, W. (1968) A research center for augmenting human intellect. *Proceedings of Fall Joint Computing Conference*, 33(1), 395-410.
3. Forlines, C. and Shen, C., DTLens: multi-user tabletop spatial data exploration. in *Proceedings of the ACM symposium on User interface software and technology*, (Seattle, WA, USA, 2005), ACM Press, 119-122.
4. Google Inc. Google Earth . http://earth.google.com/
5. Greenberg, S., Sharing views and interactions with single-user applications. in *Proceedings of the ACM.IEEE Conference on Office Information Systems*, (Cambridge, Massachusetts, USA), 227-237.
6. MapQuest. http://www.mapquest.com/
7. Microsoft. Virtual Earth. http://local.live.com/
8. Morris, M.R., Ryall, K., Shen, C., Forlines, C., and Vernier, F. Beyond "Social Protocols": Multi-User Coordination Policies for Co-located Groupware. In *Proceedings of CSCW 2004*, 262-265.
9. Roberts, J.C., On Encouraging Multiple Views for Visualization. in *Proceedings of IEEE Symposium on Information Visualization InfoVis'98*, (Research Triangle Park, NC, USA, 1998), 8-13.
10. Streitz, N., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R.., i-LAND: An interactive Landscape for Creativity and Innovation. in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, (Pittsburgh, Pennsylvania, USA, 1999), 120-127.
11. Tse, E., Shen, C., Greenberg, S., Forlines, C., Enabling Interaction with Single User Applications through Speech and Gestures on a Multi-User Tabletop. in *Proceedings of AVI 2006*, (Venezia, Italy, 2006), 336-343.