Hugin: A Framework for Awareness and Coordination in Mixed-Presence Collaborative Information Visualization

KyungTae Kim[1], Waqas Javed[1], Cary Williams[2], Niklas Elmqvist[1], Pourang Irani[2]

[1]Purdue University
West Lafayette, IN 47907-2034, USA
+1 (765) 494-0364
{kimk, wjaved, elm}@purdue.edu

[2]University of Manitoba
Winnipeg, Manitoba R3T 2N2, Canada
+1 (204) 474-8995
{umwill22, irani}@cs.umanitoba.ca

ABSTRACT

Analysts are increasingly encountering datasets that are larger and more complex than ever before. Effectively exploring such datasets requires collaboration between multiple analysts, who more often than not are distributed in time or in space. Mixed-presence groupware provide a shared workspace medium that supports this combination of colocated and distributed collaboration. However, collaborative visualization systems for such distributed settings have their own cost and are still uncommon in the visualization community. We present Hugin, a novel layer-based graphical framework for this kind of *mixed-presence* synchronous collaborative visualization over digital tabletop displays. The design of the framework focuses on issues like awareness and access control, while using information visualization for the collaborative data exploration on network-connected tabletops. To validate the usefulness of the framework, we also present examples of how Hugin can be used to implement new visualizations supporitng these collaborative mechanisms.

General terms: Design, Human Factors.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces—Graphical user interfaces (GUI)

Keywords: Collaborative information visualization, information sharing, mixed-presence visualizations.

INTRODUCTION

Real-world datasets are growing in scale, scope, and complexity. This growth requires *teams* of experts to collaborate seamlessly to analyze massive volumes of data with novel visualization and information sharing tools [19]. Collaborative visualization has been proposed as one approach to deal with this increased scale and complexity [6, 17, 18, 36, 37]. However, efficient collaboration requires flexible and powerful *coordination mechanisms* [24, 35] which are not present in most visualization systems designed to date. In particular, visualization systems and their associated interaction techniques have not been fully adapted to cope with the types of coordination mechanisms required for teams of analysts working on a problem together. This drawback is further exacerbated when groups of collaborators are distributed spatially, as is common in many real-world situations [39].

Two primary approaches exist for integrating collaborative mechanisms into information visualization. The first requires refactoring existing visualization tools to include features that facilitate collaborative work. The second approach, which we take here, is to build a general-purpose framework for integrating existing visualizations and which already supports key collaborative features such as group awareness and information access. Both approaches have tradeoffs—the former entails modifying existing tools, whereas the latter reimplementing visualizations from scratch.

To this end, we present the Hugin toolkit,[1] a framework designed to incorporate synchronous collaboration with visualization in *mixed-presence* [15, 31] settings on interactive surfaces. Mixed-presence settings are those where some participants are co-located and some are distributed (but connected by the network). Hugin provides a general platform for integrating visualization systems to create this kind of hybrid collaborative environments. While other mixed-presence collaborative systems (such as [38]) are mainly designed for general groupware applications, Hugin was developed specifically to suit tasks that are common when interacting with a visual data display, such as overview, filtering and details [30], brushing [5], and dynamic queries [29]. Additionally, Hugin builds on a large number of principles drawn from the literature in computer-supported collaborative work, such as personal territories for identity tracking and object ownership [21], information access through an object layer system [28], and awareness of other participants' workspaces through features such as telefingers and a minimap [13, 14]. These mechanisms are controlled using a personal interaction dock, unique to each individual participant.

The Hugin toolkit can be easily employed by practitioners and researchers to plug in their visualizations and get immediate access to collaborative mechanisms. To validate the

ITS'10, November 7-10, 2010, Saarbrücken, Germany.

[1]Hugin is the name of one of the principal Norse god Odin's twin ravens, Munin being the other one. Serving as Odin's eyes and ears, they leave his shoulder every day to fly all over Midgard and survey the state of the world.

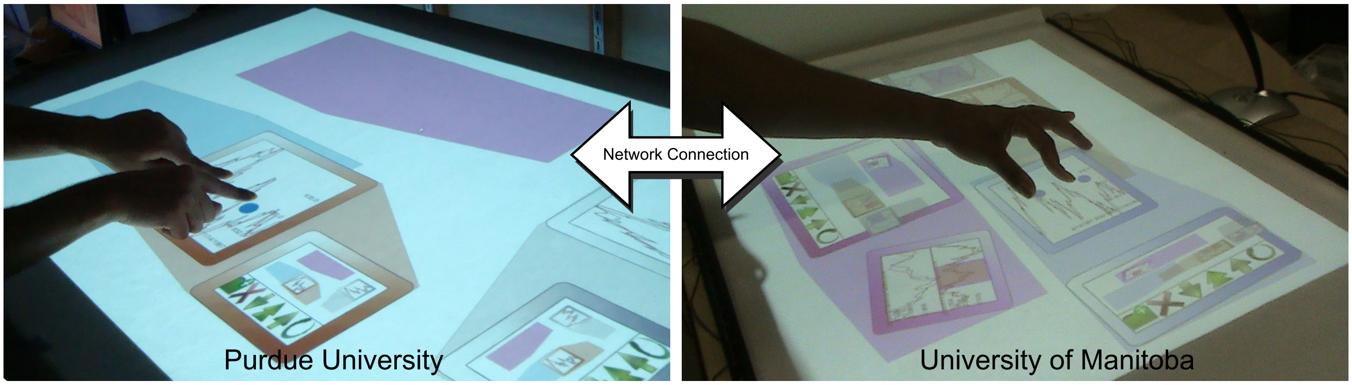


Figure 1: Users in different locations using tabletop displays running Hugin to collaboratively analyze a time-series dataset.

toolkit, we implemented three very different visualization workspaces for interactive timelines [20], scatterplots [9], and parallel coordinates [16]. We demonstrate that adding new visualizations to the framework can be done with relative ease. Furthermore, using one of these visualization workspaces, we also performed an informal user study to investigate how well Hugin supports real-time activity and how its coordination mechanisms are able to support a group of participants in a collaborative problem-solving task. The results show that participants were able to carry out typical analytical tasks with the framework and build on efforts of various team members, even when separated geographically.

Our contributions in this paper are: (1) a framework designed for building information visualization tools in mixed-presence settings; (2) a set of principles to guide the design of such a framework; and (3) a set of visualizations that have been implemented using our framework.

In the rest of this paper, we first describe the literature from which we derive our design guidelines and principles for building a framework for integrating visual tools in a mixed-presence manner. We then enumerate and motivate our design goals. We present the framework, including its features for distributed visualization, privacy, and awareness. We then describe the three visualizations we built for the toolkit. Finally, we discuss our informal user study, present the results, and conclude with a summary and future plans.

BACKGROUND

To set the stage for our work, this section discusses the literature in the intersection of computer-supported collaborative work, information visualization, and tabletop interaction.

Information Visualization

Information visualization is the graphical representation of abstract data to aid analysis, synthesis, and understanding of the data [8]. Because the data has an abstract form—for example, text, trees and graphs, and multiple dimensions—the graphical representation must be designed by the developer. Contrast this to *scientific visualization*, where the data is spatial (2D, or most frequently, 3D), and where the emphasis often is more on efficient rendering of 3D structures than information design and interaction.

Information visualization applications are characterized by a pipeline [8] that transforms data in raw form to visual representations and navigable views of these representations. Beyond standard navigation operations such as zooming and panning, visual exploration includes an array of tasks specific to visualization [40], such as overview and detail [30], queries [29], and exploration [40]. Furthermore, because visual representations are chosen by the user rather than given by the data, there is often a need for multiple linked views of the same data that can be brushed [5] across views.

For the most part, current visualization systems have all been designed for a single user. With today's massive volumes of data, information visualization tools need to embrace the technologies that allow more collaborative forms of data analysis [36]. Few visualization platforms support co-located collaboration, and even fewer target mixed-presence settings.

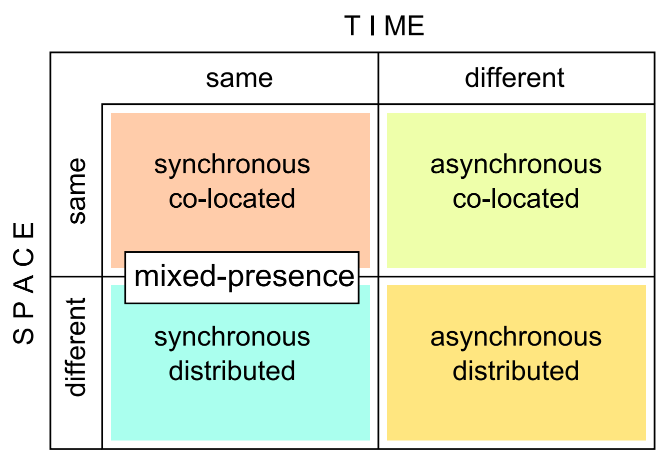


Figure 2: Space-time collaboration matrix [2] for groupware [10], including mixed-presence systems.

Collaboration and Groupware

Unlike standard applications that are primarily designed for one user that is in control of both input devices and display, *groupware* [3, 10] is computer software designed for multiple concurrent users. In computer-supported cooperative work, groupware is often classified using a space-time matrix [2, 3] (Figure 2). Broadly speaking, groupware is either synchronous or asynchronous in nature, and the collaboration takes place in the same location, or in different locations.

However, the space-time matrix can be limiting [3] because people often collaborate over the boundaries of space and time (note that we do not consider asynchronous time aspects in this paper). For the visualization field, some of the analysts with the required expertise may not be available at a single location, but may be geographically distributed. This leads to hybrid situations where some collaborators are co-located, and some are distributed. This is known as *mixed-presence* groupware [15, 31], and it has its own set of unique issues.

Collaborative Information Visualization

Over the last few decades, the most prevalent user group for visualization has been expert users working individually on a standard desktop computer. However, we are witnessing a change in user groups, shifting gradually away from focusing solely on single expert users to also include novice users as well as groups of users. Furthermore, new computer hardware like wall-sized and tabletop displays are providing exciting platforms for visualization applications to run on.

In particular, the combination of multiple users working on the same problem and large displays that facilitate collaboration [25] opens up the potential for *collaborative visualization*, a topic that has been named one of the grand challenges for visualization research [36]. This approach better models real-world problem-solving and decision-making, which more often than not are conducted in teams and not by individual analysts [19]. Early results by Mark et al. [23] for both distributed and co-located visualization systems indicate the benefits of collaboration for visualization. Balakrishnan et al. [4] show that a shared visualization significantly improves collaborative task performance.

In this paper, we concern ourselves with groupware for real-time collaborative visualization [6, 36]. Asynchronous collaborative visualization has recently become prevalent, for example on the Web for the new generation of social visualization systems like ManyEyes, Swivel, and Tableau Public. However, our target in this work is *synchronous collaboration* where a team of analysts is working together in real time on a problem. This is common for many real-world scenarios like command and control, design, and creativity.

The distributed setting has long been an active topic in scientific visualization [7, 11]. For information visualization, the U.S. Army Command Post of the Future (CPOF) [36] is a seminal example. It is also becoming a focus for the emerging field of visual analytics; for example, Brennan et al. [6] develop a distributed system for multi-user analysis.

For co-located settings, on the other hand, the Responsive Workbench [1] is a pioneering example of a visualization systems designed for a multi-user setting on a horizontal surface. Isenberg and Carpendale [17] recently presented an information visualization on tabletop displays for tree comparisons, Cambiera [18] is a tabletop visualization for document collections, and Lark [37] improves on coordination for collaboration using an embodiment of the visualization pipeline.

Coordinating Collaboration

Collaboration requires coordination to be efficient [35, 37]; without coordination, there is no collaboration, just independent efforts of multiple people working on the same problem. However, providing flexible coordination mechanisms is a challenge in itself [12], particularly for the kind of complex, large-scale, and taxing data analysis problems that collaborative visualization systems are typically applied to [19].

Most coordination mechanisms can be classified as pertaining to workspace awareness [12], territories [26], or access control [21]. Another level of complexity is added to these mechanisms for mixed-presence collaboration [39], like for our work. We discuss each of these issues below.

Awareness for Collaboration

Workspace awareness concerns the real-time understanding of the current state of the workspace and the activities of other collaborators [12]. It is vital for distributing work, avoiding duplicated efforts, and sharing information transparently between collaborators [18, 38].

Tobiasz et al. [37] use meta-visualizations to make the analysis process explicit. In their system, Lark, multiple coordinated views support participants freely switching between different collaboration styles. Cambiera [18] was designed for collaborative investigative analysis and provides workspace awareness through a technique called *collaborative brushing and linking*. However, these methods are all designed for co-located settings where the actions of other participants are visible. While many such methods would also work for mixed-presence systems, studies show that mixed-presence have unique challenges that must be considered [31].

In contrast, Tuddenham and Robinson [38] derive design guidelines for awareness in mixed-presence collaboration on tabletops through territories, orientation, and implicit communication. In later work [39], they study these effects empirically in various collaborative settings, finding differences in coordination between the settings.

Territoriality and Access Control for Collaboration

Empirical studies show that analysts working in groups often switch back and forth between individual/private and collaborative/public spaces [19, 34, 35]. Explicitly supporting this kind of private and public spaces may help awareness and coordination [37, 38], and will also support identity tracking, object ownership, and access control [21].

Agrawala et al. [1] use frame interleaving and shutter glasses on a projector-based tabletop to provide perspective-corrected and individual images of a visual workspace for two co-located collaborators. The technology can be used to create *specialized views* of information for access control.

More general work in this area [26, 35] discusses the use of personal and group territories defined by the physical seating of the group members in co-located collaboration. The concept of virtual territories is clearly important, but its generalization to the mixed-presence collaboration is not obvious. Recent work by Tuddenham and Robinson [39] shows that on mixed-presence tabletops, remote participants do not partition territories like they do in co-located collaboration.

MIXED-PRESENCE COLLABORATIVE VISUALIZATION

Given this rich literature of collaborative visualization systems and groupware in general, we derive a set of design requirements for our proposed platform for mixed-presence collaborative information visualization. These requirements are mainly inspired by the design guidelines for co-located collaborative information visualization proposed by Isenberg and Carpendale [17], as well as those for mixed-presence collaboration proposed by Tuddenham and Robinson [38]:

DR1 Mixed-presence: Many realistic data analysis tasks involve geographically distributed participants [31, 38, 39]. This is the primary design requirement of our system.

DR2 Shared interactive surface: There should be a shared interaction space [17] for the co-located participants:

- *Horizontal:* Studies show that tabletops support collaboration better than vertical displays [25, 27].
- *Large size:* The surface must be sufficiently large to accommodate multiple concurrent users [17, 38].
- *Rotatability:* Orientation is a powerful coordination mechanism [22], so the interface must support rotation and translation of graphical elements [38].
- *Simultaneous interaction:* The ability for all participants to be able to interact freely with the visualization without turn-taking is vital for effective collaboration [38].
- *Identity tracking:* Supporting roles and ownership requires identity tracking of input touches [21].

DR3 Role-based collaboration: Efficient collaboration requires supporting asymmetric expertise and authority [41].

DR4 Territory support: Territories are regions of space on the collaborative surface, and are vital for organization [26], coordination [39], and identity tracking [21].

DR5 Group awareness: Awareness of what other participants are doing is necessary for distributing labor, avoiding duplicate efforts, and implicit communication [12, 18, 37].

DR6 Information access: Collaborators should be able to control access to their views [34, 35] to avoid cluttering the shared space and create specialized views [1].

DR7 Voice communication: Voice is a ubiquitous communication channel and vital for effective collaboration [38], particularly for tabletop settings where no physical keyboard for instant messaging is readily available.

DR8 Collaboration styles: Several empirical studies point to frequent switching between loosely and closely coupled styles of collaboration [19, 34, 35], and thus an effective system must support all collaboration styles [17, 37].

THE HUGIN TOOLKIT

Hugin is an information visualization framework designed for real-time mixed-presence (DR1) collaboration over the Internet. The system is based on a client/server architecture with a single IRIS server for maintaining the shared state, and any number of IRIS clients distributed across the network (Figure 3). The framework was primarily designed for multitouch tabletop displays (DR2) because of their suitability for collaborative work [25], but because the distributed protocol is platform- and hardware-independent, Hugin clients can also be run on a standard computer using a mouse and keyboard. Thus, future extensions to the framework could support a wide array of platforms, ranging from wall-sized displays to multitouch smartphones like the iPhone.

The core functionality of the Hugin framework is distributing visualizations so that they can be used for mixed-presence collaboration. However, the design of the framework has allowed us to add built-in support for many of the coordination mechanisms described above, including role-based collaboration (DR3), territories (DR4), awareness (DR5), and privacy support (DR6). The tool currently does **not** support voice communication (DR7); instead, we employ Skype[2] for this purpose. However, by virtue of the free-form and unconstrained nature of Hugin, it does support the full spectrum of collaboration styles (DR8). We describe this in detail below.

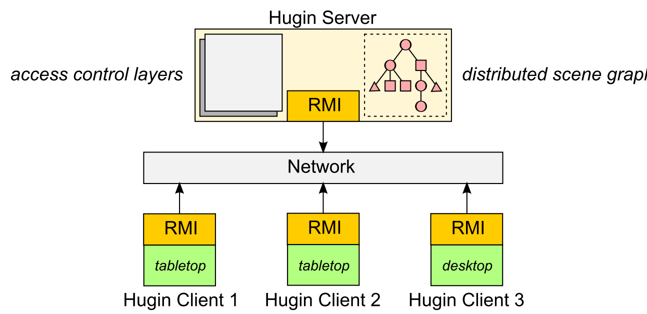


Figure 3: The Hugin distributed network architecture.

Distributed Visualization

Hugin is a client/server architecture built on top of the Java RMI (Remote Method Invocation) framework. It is based on a central *Hugin server* that manages all shared resources and coordination mechanisms, and a number of *IRIS clients* that connect to the server over the network (Figure 3). Because RMI is Java technology, the Hugin framework does require clients to run on a Java Virtual Machine, but it imposes no constraints on the actual computer platform itself.

The basic visual entity of the Hugin distributed visualization service is the *workspace* [21], a rectangular region in 2D space with an associated 3×3 affine transform. Workspaces are the Hugin equivalent of windows in standard windowing systems, and have a set of operations that can be performed on them: geometric (scale, rotate, move), input (press, tap, drag, release), access-related (acquire, release, publish, unpublish), and component-related (create, hide, delete).

To support arbitrary graphical representations, Hugin uses a distributed scene graph exported using RMI. Each workspace has an associated scene graph root. Hugin clients can populate the scene graph for a workspace by creating and adding nodes such as lines, polygons, and text to the workspace root. All connected clients will be able to access and see the shared workspaces on their local interaction space.

Managing local and remote space is an important factor for mixed-presence groupware [39]. Hugin currently overlays

[2] `http://www.skype.com/`

all workspaces on the same physical space, suitably scaled to accommodate different display geometries. In the future, we may want to extend this to other virtual arrangements of the combined interaction space, such as not overlaying client spaces but placing them side by side, overlaying only a small, shared part, or including portals into other parts of the space.

Beyond sharing graphical representations, distributed visualization applications typically also require data sharing [11]. Towards this end, Hugin supports database workspaces that can be connected to a visualization workspace to form a complete pipeline (similar to Lark [37]). This is described below.

Layer Architecture

Information access (DR6) requires the ability to create specialized views [1] of information that support independent work [34, 35]. As stated above, workspaces can both be local to an Hugin client only, or they can be shared and exported on the Hugin server so that everyone can see them. However, note that because our hardware does **not** support personalized views of the tabletop surface (like Agrawala et al. [1]), workspaces are still visible by all co-located collaborators.

To support all levels of information access, Hugin has a layer-based architecture for access control, similar to that of Shiozawa et al. [28]. All workspaces belong to one of three layers (note that personal docks are always private):

- **Private:** Local workspaces that only exist on a particular Hugin client and are thus only visible on that client (the extents of a collaborator's interaction space are visible);
- **Protected:** Shared workspaces on the server that are visible to all, but are still owned by a particular user and are thus read-only to all other participants; and
- **Public:** Shared workspaces on the server with no owner, allowing anyone to view, interact and acquire them.

The access permissions of each workspace is shown using both color and a label in the title bar of the workspace.

Figure 4 summarizes this layer structure and operations for moving between layers. These operations are invoked using the buttons on the workspace management toolbar (Figure 6).

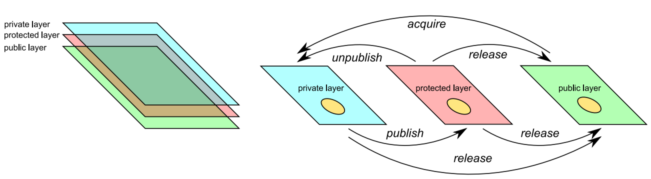
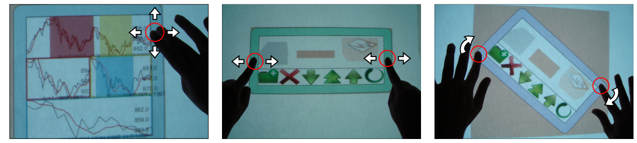


Figure 4: Hugin layer structure and layer operations.

Hugin Client Interaction

On the client side, Hugin supports the basic direct touch interaction expected from a multitouch interface (Figure 5). A workspace can be moved on the interactive surface by dragging its border, and borders can be resized by performing a two-point drag. To support rotatability [38], workspaces can be rotated by interacting with its corners.

For shared workspaces, all interactions on the workspace are translated into abstract operations that are sent to the Hugin server to be executed. For local workspaces, the operations are performed in the client itself without involving the server.



(a) Move. (b) Resize. (c) Rotate.

Figure 5: Basic multitouch gestures in Hugin.

Personal Interaction Dock

Because many tabletop displays currently do not directly support identity tracking in the hardware [21], we introduce the concept of a *personal interaction dock* to serve as the presence of a participant on the visual space. A personal interaction dock is a special type of workspace to which tools, displays, and interaction techniques can be attached. Docks are created by a simple pinch gesture in an empty space, they are color-coded, and all operations performed through a particular dock will take the identity of its owner. We rely on social protocols for users to protect their own docks [21, 24].

Figure 6 shows the standard toolbar of a personal interaction dock. This toolbar is used for all workspace management operations and can be used either in a two-step fashion, where you click the button and then the workspace to operate on, or simultaneously using a two-point multitouch gesture.

Figure 6: Workspace toolbar for a personal interaction dock. From left: create, delete, publish, release, acquire, and collect all workspaces (to within reach).

All workspaces created or acquired using a particular dock become part of the *personal interaction space* [21] associated with that dock. The personal interaction space serves as the territory mechanism [26] of Hugin (DR4) and is the combination of all workspaces (including the interaction dock) owned by the user. As stated above, the contents of local workspaces are not sent to the server, but we do send the borders of each collaborator's interaction space to indicate the presence of the user on the collaborative space. Extents are color-coded with the respective colors of each collaborator, and are computed using the convex hull of the combined workspaces belonging to the user (Figure 1 shows these convex hulls in action for two collaborating sites).

Awareness Minimap

Hugin provides an *awareness minimap* in each collaborator's interaction dock (Figure 7). The minimap is simply a bird's eye view of the whole collaborative space (or potentially both local and remote spaces for other configurations; however, Hugin currently overlays all surfaces on the same physical space). Inspired by the radar and miniature views of Gutwin et al. [13, 14], it provides the following functionality:

- **Awareness:** The minimap shows an overview of workspaces currently visible for the owner of the minimap, promoting awareness of both local and remote collaborators (DR5).
- **Telefingers:** Participants can point at any item on the minimap, causing color-coded finger blobs to show up on the interactive space at both local and remote sides. This is an extension of the telepointer concept [14], and related to recent work on arm embodiments for awareness [32, 33].
- **Reach:** Because a user can interact with workspaces in the public layer using the minimap, it solves some of the reach problems prevalent in large display interaction.
- **Orientation:** The minimap is integrated into the interaction dock, which behaves like any other workspace on the Hugin canvas. This means that users can freely rotate the minimap, allowing them to orient workspaces that are facing the wrong way on their own minimap without affecting the main canvas itself (private and protected workspaces can only be rotated by their owner, not by others).

The minimap also displays the color-coded convex hulls for the presence of each participant on the shared space.

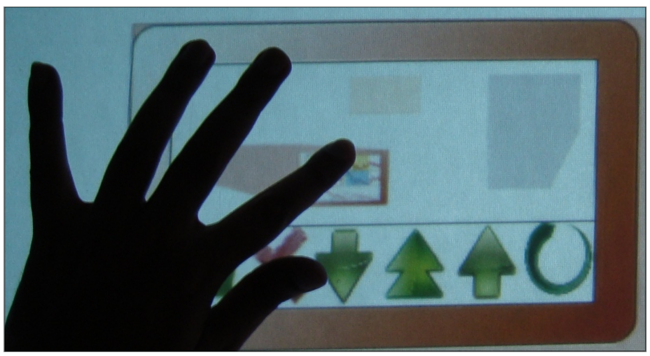


Figure 7: Interacting with the awareness minimap.

Databases, Visualizations and Pipelines

As discussed above, distributed visualization requires data-level sharing [11], and in Hugin this is accomplished using special-purpose entities called *database workspaces*. A database workspace is the visual manifestation of a dataset, and can be connected to a *visualization workspace* to form a pipeline, similar to Lark [37]. The link between a database and a visualization is shown as a visual link connecting the two workspaces on the Hugin surface (Figure 8).

In practice, users create database workspaces in Hugin using their interaction dock, and then use this database to both select which local dataset to load (from a specific directory in the Hugin distribution), as well as which visualization workspace to create and connect to the database. The framework comes with existing visualization workspaces (see the next section), but creating new workspaces using the toolkit is easy and they plug directly into the database workspace.

Note that many different visualization workspaces can be connected to the same database workspace, to either show different visual representations of the same data, or to show the same visual representation to different users. The connection between data and visualization is special because publishing a visualization may means that the connected

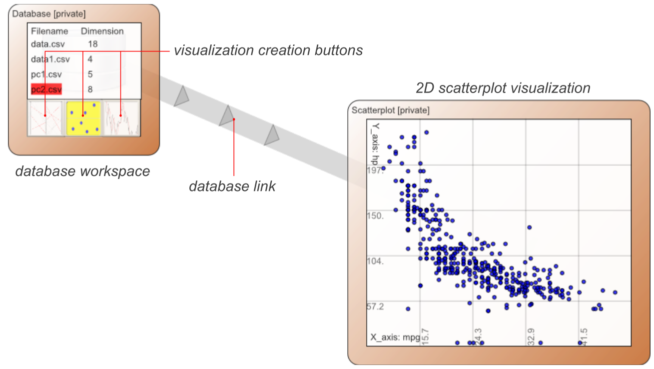


Figure 8: Database workspace connected to a 2D scatterplot for a car dataset consisting of 406 entities.

database should be made public as well. Hugin has a set of user-defined access policies for dealing with this, which includes replicating a database for when a public database with connected visualizations is acquired and made private.

IMPLEMENTATION NOTES

Hugin was implemented in Java and uses the Piccolo[3] toolkit for abstract, vector 2D graphics. The network architecture is based on RMI, the standard remote object distribution framework for Java. Because we use vector representations, even complex visualizations only require limited bandwidth.

We implemented a Piccolo2DTouch library to be able to manipulate Piccolo applications using a multi-touch tabletop display. The library uses the TUIO[4] protocol to dispatch touch events—such as the number of blobs and their position, path, and speed—to touch event handlers associated with Piccolo scene graph nodes. It is available as Open Source software on Google Code. We plan on making the Hugin toolkit similarly available as Open Source in the future.

VISUALIZATION WORKSPACES

We have built a small set of visualization workspaces to demonstrate the utility of the Hugin framework. Building a new visualization consists of the following simple steps:

1. Implement the shared visualization state as an RMI object that is stored on the server using the Hugin framework.
2. Build the visual representation using Piccolo as a scene graph subtree that will be added to the workspace.
3. Add input handlers to the scene graph to transform input events to operations on the shared state object on the server.

Because Hugin uses Piccolo for graphics, visual representations are abstract scene graphs consisting of vector shapes, text, and graphical properties. This makes it relatively easy to adapt existing visualizations built in Piccolo to a mixed-presence collaborative setting. Furthermore, Hugin manages the data flows from database workspaces, thus freeing the visualization developer from loading and transforming data.

[3] `http://www.piccolo2d.org/`
[4] `http://www.tuio.org/`

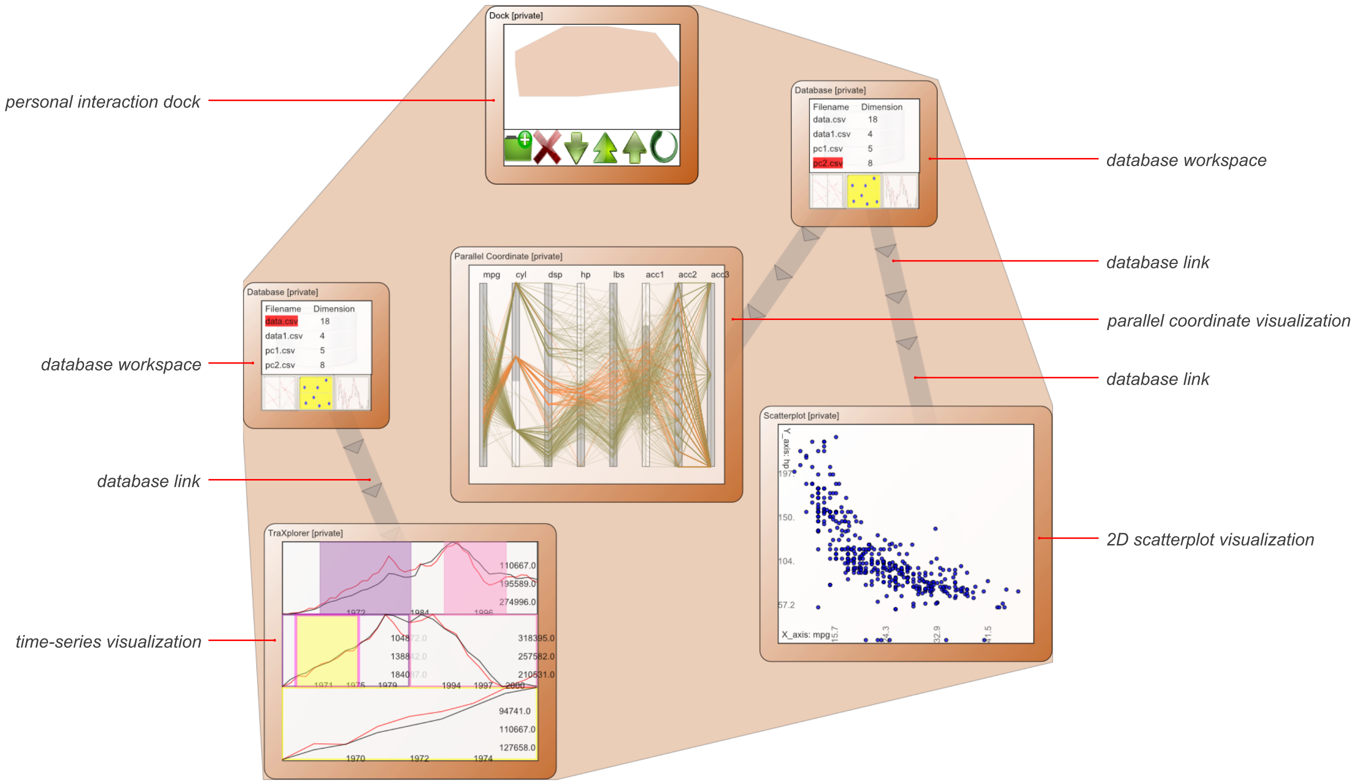


Figure 9: The personal interaction space of one user showing two database workspaces and three different visualization workspaces. Visual links connect databases and visualizations, showing what each visualization is displaying. Both the 2D scatterplot and the parallel coordinate workspace visualize the same car dataset consisting of 406 records. The time-series visualization is showing two measures of California crime data for the period 1960–2008.

Time-Series Visualization

Our time-series workspace supports a line graph representation of one or several time series, such as stock values, temperature readings, or blood pressure over time. Time is on the horizontal axis and value on the vertical.

By using the two-point selection gesture on the surface of the chart, the user can create a zoomed-in focus region that is stacked below the full chart (the original chart is resized to accommodate the new region below). This is called *stack zooming* [20]. A color-coded selection area shows the extents of each focus region. Using successive selections, the user can create a hierarchy of focus regions—see the lower left part of Figure 9, where the selection areas (purple, pink, and yellow) indicate the focused intervals of charts below.

Beyond using the two-point selection gesture, the user can also drag a selection area to pan the focus region along the time axis. Double-tapping a selection area will delete the corresponding focus region belonging to that selection area.

2D Scatterplot

We implemented a 2D scatterplot [9] workspace where the user can select which dimension in the dataset should be mapped to which axis. Scatterplots are classic statistical data graphics, assigning spatial dimensions (typically two) to actual data dimensions and plotting data records as points in the Cartesian space defined by the axes. Beyond position, scatterplots can use visual variables like the size, shape, and color of the points to convey information about the underlying data. These visual variables have limited fidelity, however, so our implementation currently does not use them.

Figure 9 (lower right) shows an example of an Hugin 2D scatterplot visualization workspace displaying two dimensions (gas mileage and horsepower) for a car dataset consisting of 406 individual entries. Beyond merely viewing the data, the workspace also supports zooming and panning navigation using standard multitouch gestures. Buttons appear when the visualization is active to allow the user to change the mapping of data dimensions to spatial axes.

Parallel Coordinates

Parallel coordinates [16] are a popular multidimensional visualization technique where spatial axes are created for each dimension in the dataset and then stacked in parallel. Data records become polylines that connect the corresponding values on each axis for that record. In this way, a trained analyst can quickly detect trends, correlations, and patterns by studying how polylines group, distribute, and connect.

Figure 10 shows a parallel coordinate visualization workspace in Hugin connected to the car dataset above. Unlike the 2D scatterplot, the parallel coordinate can display all of the dimensions in the dataset. Our implementation also supports *axis filtering* where range sliders on each dimensional axis allows for dynamic queries [29] of the data. In other words, beyond moving the thumb of the slider (representing the se-

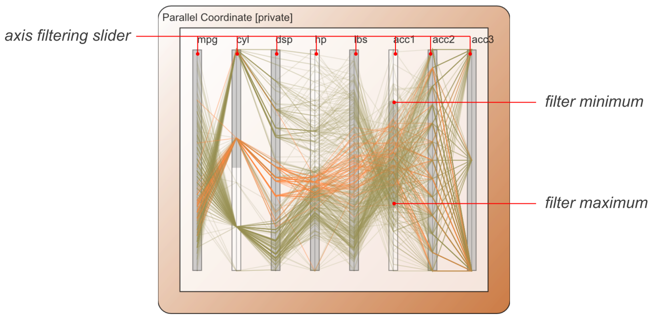


Figure 10: Detail view of the parallel coordinate visualization workspace for the 8-dimensional car dataset.

lected interval), the user can change the extents of the interval by manipulating the endpoints of the thumb. The filters for each axis are combined into a conjunction so that records that fall within all filters are colored a bright red, and those that fall outside one or more are colored a dim green.

INITIAL USER STUDY

We performed an informal user study and collected qualitative data on some of the features of the Hugin framework. The study was conducted concurrently at two different locations: Purdue University and University of Manitoba. Each site had a tabletop display, connected together using Hugin:

- **Purdue University:** an FTIR tabletop display, dimension 57” × 36” resolution 1280 × 800;
- **University of Manitoba:** a DI tabletop display, dimension 38” × 28”, resolution 1280 × 800.

Each side was also connected using a Skype voice connection with omnidirectional microphones and loudspeakers at each location. We deliberately chose to constrain the connection to audio (i.e., no video) to limit the out-of-band communication between the mixed-presence components of each quad.

We recruited 12 unpaid participants (10 male, 2 female) for this user study from the student pool at our locations (six each). We grouped participants into dyads within each site (restricted to knowing each other in beforehand), and pairs of dyads, one at each site, into quads (four in total).

Evaluation Procedure

After exposing participants on both sides to 10 minutes of supervised training, we engaged them in two temporal data analysis scenarios that took 15 minutes each:

- **Stock market index:** Participants were asked to analyze 10 years of a stock market index as well as its 20-day moving average. All participants saw the same data.
- **California crime:** Users analyzed California crime data from 1960 to 2008. Each participant was assigned a different piece of the dataset (murder, assault, theft, violence).

The task for both scenarios was to find a single time period in the given time series where a value was maximal and minimal: highest and lowest stock market value, highest and lowest crime activity across all categories of crime.

For both tasks, users were given access to the full coordination mechanisms of Hugin (except for the second task, where participants could not make their respective datasets public). After finishing all tasks, participants were asked to fill out a questionnaire about their experience, both in terms of particular Hugin features, as well as their open-ended comments. Beyond qualitative instruments, our tabletop software silently collected cinematic logs of all touches on the tabletops in both locations, and we also collected SVG screenshots of the tabletop surface at the end of each task.

Results

All groups were able to complete the tasks and provide an answer within 95% of the correct year range. While we did not record user interaction during the user study, we did observe and take notes of participant performance. These observations and the SVG screenshots indicate that presence disparity [31] was a significant factor: participants seemed more comfortable collaborating with their co-located partner. Its impact varied across sessions—in one session, one participant took the role of leader and guided the others through the tasks, whereas in another, there was very little communication between dyads until during a final consensus stage.

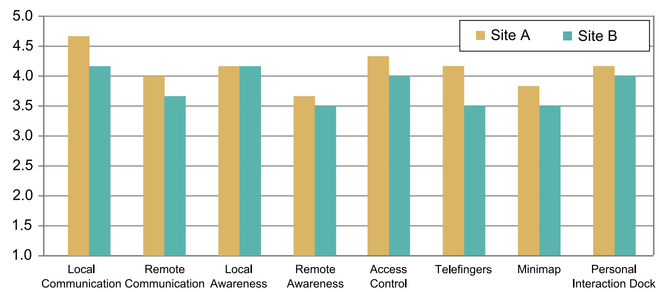


Figure 11: Average Likert scale (1-5) ratings for features of the Hugin platform, collected from our user study. Gold bars are Purdue, blue are Manitoba.

The user ratings (Figure 11) suggest that, on average, all participants felt that the features in Hugin were easy to use and accessible with little training (all rankings are above the 3 average). This is noteworthy since none of our participants were previously familiar with either remote collaboration tools, or with advanced visualization systems. Overall group and remote level of communication and awareness indicate that participants felt that both local and remote communication and awareness were well supported in Hugin.

Some participants commented on the fluidity of the environment. One participant said “with the telefingers and minimap, it felt as though the remote participants were in the room itself.” This same participant also mentioned that they felt more productive as there were fewer hands in the same locale but were able to accomplish as much as if the others were in the same place. This feedback corroborated with that of another user who mentioned that “highlighting a certain region of the visualization, making it visible to everyone, and then pointing at a certain point in the map” was some of the most useful features of Hugin. Another participant quoted the access control mechanisms as a significant feature.

Despite overall positive comments, when asked what was the more difficult aspects, one user commented that “there was no problem communicating with the local partner. But, communicating with remote partners was a bit hard, since we can see what they are doing, but not what they are focusing on." A participant also hinted at augmenting the communication with video to heighten the interactivity. When asked what features would have made collaboration easier, one participant suggested that "when I view the remote person's chart, it would be nice to see it right side up from my point of view."

Discussion

The subjective data raises some interesting points. First, the remote collaboration metrics are rated slightly lower than local collaboration. This is not unexpected because of *presence disparity* [31], where collaboration dynamics are affected by whether a participants are co-located or remote. By the same token, the concept of *display disparity* may also help explain why Manitoba participants gave slightly lower ratings than Purdue participants (less than 5% difference)—the Manitoba tabletop was smaller than the Purdue one. In other words, our results are consistent with those of Tang et al. [31].

This is also the place to begin to generalize the *characteristics* of a visual representation designed for collaboration on tabletop displays. Tobiasz et al. [37] take a step in this direction by discussing how their Lark system extends existing approaches to coordinated multiple displays (CMV) to multiple users collaborating synchronously on tabletop displays. Our results echo theirs, including the need for embedding interactions, keeping visuals minimal, and supporting spatial and temporal flexibility in the collaboration. In addition to these, we would add scale-independent visual representations, role-based interaction, and the capability for specialized views.

Finally, we chose Skype to realize the DR7 requirement for voice support, but we found that not integrating this mechanism into the platform seriously limited its use. In our experiment, the voice became a separate channel, whereas for an integrated voice mechanism, we could have augmented it with visual cues to help participants see who is talking, as well as allow them to indicate to whom they are talking.

CONCLUSION AND FUTURE WORK

We have presented Hugin, a visualization framework for synchronous collaboration on tabletops where the analysts are distributed on the network. Our contributions are as follows:

- A software platform (Hugin) for mixed-presence collaborative visualization on tabletop displays over the Internet;
- Visualizations implemented in Hugin, including time-series charts, scatterplots, and parallel coordinates; and
- Results from an informal user study exploring the general utility and usability of the framework.

The outlook for collaborative visualization is promising, and we have only begun to scratch the surface of this exciting topic. In the future, we plan on exploring additional methods for control, coordination, and awareness, mechanisms that clearly are instrumental to effective and compelling collaboration—co-located, distributed, and mixed alike.

ACKNOWLEDGMENTS

This work was partially supported by the U.S. Department of Homeland Security's VACCINE Center under award no. 2009-ST-061-CI0001. Thanks to the reviewers for feedback.

REFERENCES

1. M. Agrawala, A. C. Beers, B. Fröhlich, I. McDowall, P. Hanrahan, and M. Bolas. The two-user responsive workbench: Support for collaboration through individual views of a shared space. In *Proceedings of SIGGRAPH*, 327–332, 1997.

2. L. M. Applegate. Technology support for cooperative work: A framework for studying introduction and assimilation in organizations. *Journal of Organizational Computing*, 1(1):11–39, 1991.

3. R. M. Baecker. *Readings in Groupware and Computer-Supported Cooperative Work*. Morgan Kaufmann Publishers, 1993.

4. A. D. Balakrishnan, S. R. Fussell, and S. B. Kiesler. Do visualizations improve synchronous remote collaboration? In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 1227–1236, 2008.

5. R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

6. S. E. Brennan, K. Mueller, G. Zelinsky, I. Ramakrishnan, D. S. Warren, and A. Kaufman. Toward a multi-analyst, collaborative framework for visual analytics. In *Proceedings of the IEEE Symposium on Visual Analytics Science & Technology*, 129–136, 2006.

7. K. W. Brodlie, D. A. Duce, J. R. Gallop, J. P. R. B. Walton, and J. D. Wood. Distributed and collaborative visualization. *Computer Graphics Forum*, 23(2):223–251, 2004.

8. S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in information visualization: Using vision to think*. Morgan Kaufmann Publishers, 1999.

9. W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, Sept. 1984.

10. S. Greenberg and D. Marwood. Real time groupware as a distributed system: Concurrency control and its effect on the interface. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, 207–217, 1994.

11. I. J. Grimstead, D. W. Walker, and N. J. Avis. Collaborative visualization: A review and taxonomy. In *Proceedings of the Symposium on Distributed Simulation and Real-Time Applications*, 61–69, 2005.

12. C. Gutwin and S. Greenberg. Design for individuals, design for groups: Tradeoffs between power and workspace awareness. In *Proceedings of the ACM Conference on Computer-Supported Collaborative Work*, 207–216, 1998.

13. C. Gutwin and S. Greenberg. Effects of awareness support on groupware usability. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, 511–518, 1998.

14. C. Gutwin, M. Roseman, and S. Greenberg. A usability study of awareness widgets in a shared workspace groupware system. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, 258–267, 1996.

15. P. Hutterer, B. Close, and B. H. Thomas. Supporting mixed presence groupware in tabletop applications. In *Proceedings of IEEE Tabletop*, 63–70, 2006.

16. A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

17. P. Isenberg and S. Carpendale. Interactive tree comparison for co-located collaborative information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1232–1239, 2007.

18. P. Isenberg and D. Fisher. Collaborative brushing and linking for co-located visual analytics of document collections. *Computer Graphics Forum*, 28(3):1031–1038, 2009.

19. P. Isenberg, A. Tang, and M. S. T. Carpendale. An exploratory study of visual information analysis. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, 1217–1226, 2008.

20. W. Javed and N. Elmqvist. Stack zooming for multi-focus interaction in time-series data visualization. In *Proceedings of the IEEE Pacific Visualization Symposium*, 33–40, 2010.

21. K. Kim, T. D. Kulkarni, and N. Elmqvist. Interaction workspaces: Identity tracking for multi-user collaboration on camera-based multi-touch tabletops. In *Proceedings of the IEEE VisWeek Workshop on Collaborative Visualization on Interactive Surfaces*, 2009.

22. R. Kruger, S. Carpendale, S. D. Scott, and S. Greenberg. Roles of orientation in tabletop collaboration: Comprehension, coordination and communication. *Computer Supported Cooperative Work*, 13(5/6):501–537, 2004.

23. G. Mark, K. Carpenter, and A. Kobsa. A model of synchronous collaborative information visualization. In *Proceedings of the International Conference on Information Visualization*, 373–383, 2003.

24. M. R. Morris, K. Ryall, C. Shen, C. Forlines, and F. Vernier. Beyond 'social protocols': multi-user coordination policies for co-located groupware. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 262–265, 2004.

25. Y. Rogers and S. E. Lindley. Collaborating around vertical and horizontal large interactive displays: which way is best? *Interacting with Computers*, 16(6):1133–1152, 2004.

26. S. D. Scott, M. S. T. Carpendale, and K. M. Inkpen. Territoriality in collaborative tabletop workspaces. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 294–303, 2004.

27. S. D. Scott, K. D. Grant, and R. L. Mandryk. System guidelines for co-located, collaborative work on a tabletop display. In *Proceedings of the European Conference on Computer Supported Cooperative Work*, 159–178, 2003.

28. H. Shiozawa, K. ichi Okada, and Y. Matsushita. Perspective layered visualization of collaborative workspaces. In *Proceedings of the ACM Conference on Supporting Group Work*, 71–80, 1999.

29. B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, Nov. 1994.

30. B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, 336–343, 1996.

31. A. Tang, M. Boyle, and S. Greenberg. Understanding and mitigating display and presence disparity in mixed presence groupware. *Journal of Research and Practice in Information Technology*, 37(2):71–88, 2005.

32. A. Tang, C. Neustaedter, and S. Greenberg. VideoArms: embodiments for mixed presence groupware. In *Proceedings of the British HCI Conference*, volume 20, 85–102, 2006.

33. A. Tang, M. Pahud, K. Inkpen, H. Benko, J. C. Tang, and B. Buxton. Three's company: understanding communication channels in three-way distributed collaboration. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 271–280, 2010.

34. A. Tang, M. Tory, B. Po, P. Neumann, and S. Carpendale. Collaborative coupling over tabletop displays. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, 1181–1190, 2006.

35. J. C. Tang. Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies*, 34(2):143–160, 1991.

36. J. J. Thomas and K. A. Cook, editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society, 2005.

37. M. Tobiasz, P. Isenberg, and S. Carpendale. Lark: Coordinating co-located collaboration with information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1065–1072, 2009.

38. P. Tuddenham and P. Robinson. Distributed tabletops: Supporting remote and mixed-presence tabletop collaboration. In *Proceedings of IEEE Tabletop*, 19–26, 2007.

39. P. Tuddenham and P. Robinson. Territorial coordination and workspace awareness in remote tabletop collaboration. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2139–2148, 2009.

40. J. S. Yi, Y. ah Kang, J. T. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.

41. H. Zhu. From WYSIWIS to WISINWIS: role-based collaboration. In *Proceedings of the IEEE Conference on Systems, Man & Cybernetics*, 5441–5446, 2004.