A Widget Framework for Augmented Interaction in SCAPE

Leonard D Brown  
Beckman Institute  
University of Illinois  
Urbana, IL 61801  
lbrown2@uiuc.edu

Hong Hua  
Information & Computer Sciences  
University of Hawaii  
Honolulu, HI 96822  
hhua@hawaii.edu

Chunyu Gao  
Beckman Institute  
University of Illinois  
Urbana, IL 61801  
cgao@uiuc.edu

ABSTRACT
We have previously developed a collaborative infrastructure called SCAPE – an acronym for Stereoscopic Collaboration in Augmented and Projective Environments – that integrates the traditionally separate paradigms of virtual and augmented reality. In this paper, we extend SCAPE by formalizing its underlying mathematical framework and detailing three augmented Widgets constructed via this framework: CoCylinder, Magnifier, and CoCube. These devices promote intuitive ways of selecting, examining, and sharing synthetic objects, and retrieving associated documentary text. Finally we present a testbed application to showcase SCAPE’s capabilities for interaction in large, augmented virtual environments.

KEYWORDS: Tangible User Interface (TUI), Human Computer Interaction (HCI), Virtual Reality (VR), Augmented Reality (AR), Head-Mounted Display (HMD), Head-Mounted Projective Display (HMPD).

1 INTRODUCTION
An ideal ubiquitous environment should allow users to collaborate via intelligent, intuitive interfaces without being restricted by the complexity of the computing infrastructure. Such non-traditional computing environments have gained significant popularity in recent years. Subsequently, researchers in the areas of computer-supported cooperative work (CSCW) and human-computer interaction (HCI) have devoted efforts to the design of new infrastructures for multi-user interaction and to the construction of less cumbersome, non-traditional interface modalities.

1.1 Related Work
A large body of CSCW applications involves Virtual Reality (VR), which provides an interactive workspace consisting entirely of synthetic graphical and possibly aural elements. Applications of VR commonly use Head-Mounted Displays (HMDs) [25] and projection-based systems such as CAVE Automated Virtual Environment [8]. In an alternative use of projection-based technology, Kröger et al. [20] introduced the task-oriented, semi-immersive Responsive Workbench, and Agrawala et al. [1] extended it to dual independent viewpoints. In these systems, virtual images are projected only onto the top of the workbench, thus retaining the innate visual acuity of the rest of the physical environment. Kitamura et al. [19] proposed the IllusionHole as another projection-based, semi-immersive system. IllusionHole permits three or more views of a virtual object by using an occluding mask to obscure each user’s display surface from others.

Of great interest are systems that extend the VR paradigm by integrating physical objects into the virtual workspace, thus minimizing the cognitive seams between real and synthetic. For example, Szalavari et al. [29] and Billinghamurst and Kato [6] proposed using see-through HMDs in a collaborative interface permitting multiple local or remote users to work in an augmented, semi-immersive world. The Virtual Showcase [4] alternatively uses a stationary projection screen and convex assembly of beamsplitters to make a 3D virtual object appear encased within a conical enclosure.
A challenging extension of Augmented Reality is the integration of virtual and real components into cohesive tools that can be used to directly manipulate the augmented dataset; such tools are known as “tangible” devices [16]. Early work toward tangible interaction includes the “Worlds in Miniature” (WIM) navigation tool [27], which registers a virtual floorplan of an immersive environment with a clipboard held by the user; the user can subsequently specify a position on the hand-held WIM to jump between contexts in the immersive world.

Tangible devices are among the most exciting advances yet toward truly ubiquitous computing; however they also raise new questions as to what constitutes an “intelligent” or “intuitive” tool. Researchers have diverged somewhat in their solutions to this problem, with some concentrating on task-specific constructs and others considering generic, multi-modal approaches.

Observing a task-oriented approach, Anderson et al. [2] developed an architectural modeling application permitting construction of 3D virtual buildings via physical blocks. Piper et al. [22] used a semi-immersive, projective workbench for Illuminating Clay, an augmented landscape analysis package. Users of this system deform a white clay miniature landscape, viewing in real-time the changes to virtual topographic data superimposed onto it. Camarata et al. [7] presented Navigational Blocks consisting of four haptic cubes whose orientation and relative positions control a historical database of textual or graphical exhibits on a kiosk monitor. The cubes provide three distinct modes of interaction that allow: selection of an exhibit category (Who, What, Where, and When), navigation within that category, and haptic feedback about the Boolean relationships between items selected by the cubes.

Further, Hinckley et al. designed a set of two-handed, neurosurgical imaging tools that rely on commonly used surgical interaction metaphors [10]; the use of such pre-existing metaphors can substantially reduce training time while increasing the tools’ intuitiveness for their intended users. While the application domain here is medical analysis, it is not difficult to imagine how these devices could be generalized to many other contexts also.

Several tangible interfaces have been specifically engineered to be reusable and multi-modal. These include metaDESK [30] and Contextualized Open Hypermedia [26]. The metaDESK system formalizes the relationship between the graphical user interface (GUI) and the tangible user interface (TUI), providing a variety of generalized, physical “phicons,” “phandles,” and lenses for augmented manipulation; the Contextual Open Hypermedia system models its interface around everyday interaction metaphors, hence providing tools for “sprinkling,” “dipping,” “holding,” and “selecting” objects.

Among the most robust multi-modal constructs, Rekimoto et al. presented the DataTile system [23], which consists of a set of modular, transparent tiles that can be placed arbitrarily onto a projective workbench and then manipulated graphically via a stylus. The tiles are arranged dynamically by users to produce arbitrarily complex behaviors, synonymous to the way complex algorithms are formed from simpler operators.

1.2 Overview
Both the VR and AR infrastructures above typically address interaction from a perspective that is either egocentric (“inside-out”) or exocentric (“outside-in”). The former category includes immersive displays such as CAVE [8], while the latter category consists of semi-immersive systems like the Responsive Workbench [20]. We have developed a shared workspace called SCAPE – an acronym for Stereoscopic Collaboration in Augmented and Projective Environments – that merges a semi-immersive, augmented workbench with a corresponding room-sized, immersive environment [11].

In our continuing development of SCAPE, our principal design goals include the following: (1) the simultaneous dual-mode visualization of a dataset from both egocentric and exocentric perspectives; (2) a set of intelligent interface mechanisms allowing users to maneuver, coordinate, and relate information within this dual-mode environment; and (3) a set of generic and multi-modal interfaces permitting tangible selection, examination, and sharing of arbitrary virtual information.

In this article, we move toward these objectives by enhancing SCAPE with an augmented interface framework. More specifically, we: (1) succinctly formalize the transformation hierarchy governing interaction in SCAPE; (2) develop augmented devices via this hierarchy permitting users to intuitively select, examine, and perceptually coordinate information in the two scales of visualization; and (3) showcase SCAPE’s potential for interaction via a testbed application.

Figure 2: Head-Mounted Projective Display (HMPD). (a) A conceptual illustration. (b) A compact prototype.
In the remainder of this article, we summarize the existing SCAPE system in Section 2 and define the mathematics of interaction in Section 3. In Section 4, we extend SCAPE with three augmented interface devices: CoCylinder, Magnifier, and CoCube. In Section 5, we present an application, Aztec Explorer.

2 SCAPE: A COLLABORATIVE INFRASTRUCTURE

We now summarize the SCAPE system infrastructure [11], which serves as the foundation for our Widget framework. We provide a synopsis of the Head-Mounted Projective Display (HMPD) technology on which SCAPE is based and describe the basic components of SCAPE’s infrastructure.

2.1 Overview of HMPD Technology

SCAPE is based upon emerging Head-Mounted Projective Display (HMPD) technology [18], which lies on the boundary between conventional HMDs and CAVE-like projective displays. Conceptually, an image on a miniature display (LCD) is projected through a beamsplitter into object space, where it is reflected off a retro-reflective screen in the environment back to the exit pupil (Fig 2-a). Due to the essence of retro-reflectivity, a ray hitting the screen is ideally reflected back on itself regardless of its incident angle with the screen surface. This property allows virtual images projected from the HMPD to be viewed on the screen from arbitrary angles; it also enables a shared workspace in which each user views the synthetic dataset from his or her own unique perspective.

With its large field of view (FOV), lightweight and low distortion optics, and innate occlusion of virtual objects by real objects, the HMPD can yield effective visualization capabilities [13,14,17]. Hua et al. [12,15] has developed a compact, light-weight HMPD prototype (750 grams) that uses custom-designed lenses permitting 52° FOV. An anterior view of this prototype is displayed in Fig 2-b. The HMPD has been demonstrated as an alternative solution for a wide-range of augmented tasks.

In our SCAPE implementation [11], the workbench is placed in the center of the room and has a retro-reflective surface of 3x5 feet. The room is a four-walled cage composed of eight segments: four 6-foot flat walls and four arced corners with 3-foot radii. The 9-foot tall wall and corner segments are assembled before being coated with retro-reflective film, and one arced corner serves as a hinged door. We further employ a set of generic devices to manipulate and track objects within the environment. For example, we use a HiBall3000 optical sensor by 3rdTech [www.3rdtech.com] for precise head tracking, a Flock of Birds magnetic sensor by Ascension Tech. for limb tracking, and a 5DT DataGlove [www.5dt.com] for hand gesture recognition. We have also developed a simple vision-based 2D tracking system to recognize physical objects placed on the workbench [13].

3 AN INTERFACE FRAMEWORK

For many existing infrastructures, the fundamental coordinate system hierarchy is either scarcely documented or difficult to extend for complex augmented devices. In this section, we concisely identify SCAPE’s transformation hierarchy in the hope that it promotes multi-modal, augmented interaction, which is the focus of this paper.

As discussed, SCAPE integrates two scales of visualization, including the augmented workbench view and the large-scale immersive view on the room display. In the former view, absolute sensor measurements are required in one-to-
one physical scale. For the latter view, relative or scaled sensor measurements are needed so that users are not confined to the physical extents of the room. SCAPE’s transformation hierarchy serves as the mechanism for fusing these views (Fig. 4). At the root of this hierarchy exists the virtual World Global coordinate system, and the egocentric environment, or macro-scene, resides in $W_{\text{Macro}}$ with subsequent transform $T_{W_\text{E} \leftarrow \text{Macro}}$.

### 3.1 User Mechanics

Within the World Global context, we specify a World Local ($W_L$) system serving as the working context of all users, each labeled Actor'. The World Local system corresponds to the physical extents of the SCAPE room and may be transformed arbitrarily within $W$ via each Actor’s instance of $T_{W_\text{E} \leftarrow W_i}$. The Actor’ can thus navigate virtual spaces that are significantly larger than the physical extents of the room by transforming $W_L$ (Fig. 5). This can be accomplished with an interface device such as a data glove, six degrees-of-freedom mouse, or the augmented markers discussed in Section 5.1.

The user’s absolute location within the SCAPE room, or head position $W_H^i$, is then specified relative to the World Local frame via $T_{W_\text{E} \leftarrow H}^i$ (Fig. 6-a). We assume that a tracking sensor specifies the Actor’ head location relative to the tracker transmitter, $T_{T \leftarrow S}^i$, and that empirically-defined static transforms appropriately align the head with the sensor coordinate system, $T_{S \leftarrow H}^i$, and the tracker transmitter with the World Local coordinate system, $T_{W_\text{E} \leftarrow T}^i$. The head’s composite transform is thus

$$T_{W_\text{E} \leftarrow H}^i = T_{W_\text{E} \leftarrow T}^i T_{T \leftarrow S}^i T_{S \leftarrow H}^i.$$  

Here we assume that the head sensors of all Actors are controlled by a singular tracking device, yielding one $T_{W_\text{E} \leftarrow T}^i$. Multiple transforms $T_{W_\text{E} \leftarrow T}^i$ are needed if this is not the case. Other extremities for Actor’, such as hands or legs, may be tracked similarly and possess synonymous transformation hierarchies.

We must also compensate for the individual transforms from each eye to the head sensor, as ignoring these offsets yields registration and perception discrepancies [5]. We hence add a head to sensor transform $T_{STT}^i$ and separate head-relative coordinate systems for each eye, yielding the hierarchy illustrated in Fig. 6-a. To achieve accurate registration of real and virtual objects, precise calibration of static transforms $T_{W_\text{E} \leftarrow T}^i$, $T_{S \leftarrow H}^i$, $T_{H \leftarrow \text{Left}}^i$, and $T_{H \leftarrow \text{Right}}^i$ is also required; these transformations are typically estimated through a nontrivial calibration process that must be applied to each display device [9].

### 3.2 Device Mechanics

The transformations of arbitrary physical objects are likewise defined relative to the World Local system. Arbitrary physical objects are sensibly specified relative to the room’s origin such that $T_{W_\text{E} \leftarrow \text{Obj}}$. This transformation is identical for all local users since they share the same physical workspace; a physical object remains in the same position relative to all Actors even as they move disjointly within the larger World Global context.

We can consider the SCAPE workbench transform synonymously, designating it as $T_{W_\text{E} \leftarrow \text{B}}$. For intuitive convenience, the micro-scene is then specified in a coordinate system relative to workbench space $W_B$ (Fig. 4). We define $T_{B \leftarrow \text{Micro}}$ such that the transformation to the World Local frame is

$$T_{W_\text{E} \leftarrow \text{Micro}} = T_{W_\text{E} \leftarrow \text{B}} T_{B \leftarrow \text{Micro}}.$$

![Diagram](image-url)  

Figure 6: Details of the SCAPE transform hierarchy. (a) Transform for Actor’s eyes. (b) Transforms for room-relative (j) and bench-relative (k) Devices.
Augmented interface devices, including the Widgets discussed in Section 4, are then defined as being either room-relative ($D_R^j$) or workbench-relative ($D_B^k$), depending on their usage. These Widgets employ secondary tracking devices with static room-relative and bench-relative offsets $T_{W_j}^{-T'}$ and $T_{B_k}^{-T''}$, respectively. Again we assume that the sensors of room-relative devices are controlled by a singular tracking device, yielding one $T_{W_j}^{-T'}$; similar reasoning applies to $T_{B_k}^{-T''}$. The complete transform hierarchies are shown in Fig. 6-b.

3.3 Addressing Data Privacy

For security reasons, we cannot presume symmetric access of all users to data in the environment. As each user wears his or her own projective display, HMPD is innately suited to the viewing of customized virtual data, such that providing user-privileged access is a trivial process. As one privacy construct, we limit the accessibility of Widgets by constraining their ownership to particular Actors, such that the Widget is essentially “turned off” for users not owning it. A subset of Actors owning the same augmented device can then confer as a group independent of the larger Actor community. If the Widget also maintains a list of its owners, it may identify and interface intelligently with each particular Actor, restoring unique saved state or preferences from previous encounters. Some possibilities of Widget interaction are explored further in the next section.

4 WIDGET INTERFACE MODALITIES

We now discuss three augmented Widgets and present the less trivial details of their implementation, particularly relating to the framework of Section 3. Used in conjunction with the interactive workbench and room display, these devices provide intuitive ways of examining and coordinating data presented by SCAPE’s two modes of visualization. The Widgets include the CoCylinder, Magnifier, and CoCube.

4.1 CoCylinder Widget

As a simple test of the Widget framework, we have constructed a large cylindrical device for the viewing of life-size objects. The CoCylinder is a 360° retro-reflective surface into which a life-size virtual object is projected (Fig. 7-a). Collaborators can thus encircle the display, allowing simultaneous observation of the virtual object from unique perspectives and face-to-face conferral with other users.

For our prototype, we have constructed a cylinder 48 inches tall with a diameter of 15 inches, and coated it in retro-reflective film. To the base of the display, we have additionally attached a rotation stage with an Ascension Tech. Flock of Birds magnetic sensor; this assembly records the display’s azimuth rotation. Collaborators can additionally examine the encapsulated virtual object from varying perspectives by physically rotating the augmented device in place (Fig. 7-b).

This configuration is functionally similar to the Virtual Showcase [4] and IllusionHole [19] setups but requires significantly less computational overhead and can be trivially extended to other viewing volumes. We can construct other arbitrary volumes (see Section 4.3) by simply coating a selected physical object with retro-reflective material and matching the virtual object’s size with that of its physical counterpart. Further, the CoCylinder design allows tangible interaction with the physical device to rotate the encapsulated object, a feature that is more problematic to implement in other systems.

4.2 Magnifier Widget

The Magnifier is a more robust augmented tool allowing closer inspection of virtual data on the interactive workbench and the perceptual coordination of the micro- and macro-scene visualizations. The hardware consists of a passive, mirror-like prop with a framed, retro-reflective window of 3x5 inches and an attached Flock of Birds sensor (Fig. 1-a). We assume that the workbench’s micro-scene cannot natively display intricate details, because its visualization has been down-scaled significantly or is of a much lower level-of-detail than the corresponding macro-scene. We hence associate the Magnifier’s virtual inset camera with the high detail macro-scene such that, as the Magnifier is moved over virtual objects on the workbench, corresponding enlargements are projected onto the Magnifier’s retro-reflective window (Figs. 1,8).
Implementing a Magnification Algorithm. The Magnifier requires a four-step, multi-pass rendering algorithm. As a first step, we specify the region of the drawing buffer subtended by the device’s framed, retro-reflective screen. This is accomplished by enabling a stencil-buffer test and determining screen placement via the bench-relative transform $T_{W_{Mag},L}$. We then define a silhouette by drawing the area of its retro-reflective screen into the stencil buffer. To determine the contents of the virtual inset requires three additional steps. We must first determine the centroid of the Magnifier’s virtual inset relative to the macro-scene (Fig. 9). This calculation requires the transformation from the Magnifier to the macro-scene, which is defined as

$$T_{k}^{Mag} = S_{Macro-Micro} T_{Micro-B} T_{Bk-D}.$$ 

This chain is derived from transforms that are readily available in the interface framework, provided we note that $S_{Macro-Micro}$ is the scaling matrix from micr-scene to macro-scene. In the second step, we position the Widget’s magnifying camera ($Cam'$) above the macro-scene by examining the location of the Actor relative to the workbench and thus to the Magnifier device (Fig. 9). The transformation from the Actor’s left eye to the Magnifier serves as the offset of $Cam'$ from the inset window in macro-scene space, as follows:

$$T_{i,k}^{Mag-Cam'} = T_{DG-Micro} T_{Micro-Left} = T_{k}^{Mag-Micro} T_{Micro-B} T_{Bk-W_{Mag}} T_{W_{Mag},L}.$$ 

Thus the transformation of $Cam'$ in the macro-scene is

$$T_{k}^{Macro-Cam'} = T_{k}^{Macro-Mag} T_{Mag-Cam'}.$$ 

Finally, we compute the magnifying camera’s view direction through the virtual inset centroid, and with a sufficiently narrow field-of-view, the perceived image will appear magnified in the Widget’s inset window. The rendering algorithm is summarized concisely by the following pseudocode:

```plaintext
method RenderInset()

    MagW, TComposite ← T_{ik}^{k} W_{-D}
    Cam'. TComposite ← T_{ik}^{k} Macro-Cam'
    Cam'. V_{gaze} ← (T_{ik}^{k} Macro-Mag V_{Inset}) − (T_{ik}^{k} Macro-Cam' V_{From})
    Stencil.Enable()
    MagW.Traverse()
    Stencil.Disable()
    Cam'. Action()
    Universe.Traverse()
```

4.3 CoCube Widget

Providing the most significant 3D augmentation of our prototype Widgets, the CoCube allows for intelligent selection and examination of very large or distant objects in the immersive macro-scene (Fig. 10-b), and for the query of supplementary textual data related to the selected objects (Fig. 11). The Widget’s hardware consists of a handheld, 10-inch cube coated in retro-reflective film, with framed, diffusing edges accentuating the reflective viewing surfaces (Fig. 10-a). Attached to the inside of the device is a Flock of Birds sensor.

The CoCube has three distinct modes of interaction: selection, inspection, and cubic book. In selection mode, the CoCube allows a user to “capture” virtual objects from the surrounding macro-scene. The closest user automatically engages selection mode when he or she holds the CoCube at eye level in his or her line-of-sight. The closest object in the surrounding virtual environment is then minified to the dimensions of the CoCube’s physical box, and the Widget shifts to inspection mode (Fig. 11-a). The user can hence manipulate the object like a 3D model in a transparent box, providing a unique, exocentric view of the model at a higher level-of-detail than its counterpart on the interactive workbench.
Once the user has captured an object, he or she can also toggle to cubic book mode to access and optionally bookmark accompanying documentary text. In our preliminary interface, we map 3D, multi-faceted pages around the CoCube such that the user can flip through pages by continuously rolling the CoCube in the given certain direction; pitching upward advances to succeeding pages, while pitching downward moves to preceding pages.

An individual page consists of three display surfaces segmenting the information by type. The primary display surface contains body text (Fig. 11-b), and perpendicular to this are left and right auxiliary surfaces; the left face presents footnotes and references (Fig. 11-c), while the right face contains illustrations and diagrams (Fig. 11-d).

We have chosen this segmented page construction for the following reasons: (1) the HMPD’s native display resolution limits the amount of text that can be shown concurrently, particularly since the projected area of the CoCube’s viewing surface is significantly less than the total LCD display area under normal viewing conditions; (2) studies indicate user access to data tends to exhibit locality of reference and focus [24], which we hence extend to a textual interface by grouping similar information into the same volumetric locations; and (3) the captured object’s textual database must be self-contained within the CoCube to promote data privacy and to simplify its transport within the workspace.

The CoCube’s functionality also extends to collaborative settings. Similar to the CoCylinder, this Widget allows multiple users to observe the same cubic display via their own unique perspectives. As discussed in Section 3.1, the virtual workspaces of multiple users do not necessarily overlap, as the users could be exploring disjoint areas of the macro-scene. Consequently the CoCube can be used as a coordination tool to relay information about a user’s unique view in the macro-scene to his or her collaborators; the user first captures an object into the CoCube from the macro-scene and then passes that object, via the CoCube, to collaborators exploring elsewhere. The user can also toggle to cubic book mode and bookmark a particular page of text for discussion.

The user can hence facilitate group interaction by either holding the CoCube while directing a discussion or passing the Widget individually among his or her collaborators. Further, the collaborators can do likewise, capturing objects into the CoCube from their views and passing it back to the user. We note that only users having access privileges to the CoCube can actually view and manipulate its virtual contents (Section 3.3).

**Implementing a Selection Algorithm.** We use a set of heuristics to determine if the CoCube’s selection mode should be activated. We initially find the closest Actor with ownership privileges, marking him or her as the controlling Actor $i$. We then check if the CoCube’s height above the floor is near that of the Actor $i$’s eye, and if its centroid is in the Actor $i$’s line-of-sight to within some activation angle tolerance $\varepsilon$. The latter heuristic yields a simple dot product inequality,

$$v_{\text{gaze}} \cdot v_{\text{actor-cube}} > 1 - \varepsilon \cdot$$

Provided that the heuristics are met, the CoCube casts a ray along the Actor $i$’s line-of-sight into the macro-scene, finding the closest intersecting object, which is subsequently minified into the CoCube. This calculation requires the eye to macro-scene transform $T^{i}_{\text{Macro-Left}} = T^{i}_{\text{Macro-Right}} T^{i}_{W+\text{W}L} T^{i}_{\text{W}} T^{i}_{T} T^{i}_{H} T^{i}_{\text{Left}}$. 

![Figure 10: The CoCube Widget. (a) Manipulating the physical component. (b) The augmented visualization of a temple captured from Macro-scene.](image)

![Figure 11: The CoCube cubic book mode. (a) A temple captured from Macro-scene. (b) An associated page of body text. (c) The left auxiliary face with references. (d) The right auxiliary face with diagram and caption.](image)
5 TESTBED APPLICATION: AZTEC EXPLORER

In this section, we explore SCAPE’s interfaces in the context of a synthetic environment that is disorientingly large. Our testbed, Aztec Explorer, features a scale model of Tenochtitlan, an ancient Aztec city. We have enhanced a freeware mesh from 3DCAFE [www.3dcafe.com] and created multiple levels-of-detail. Visualized in SCAPE’s room display is the full-scale, high-detail city measuring two kilometers across. On the interactive workbench, we project a corresponding low-resolution map, at approximately 1:2800th scale, through which users can find their orientation in the macro-scene.

5.1 Navigating the Macro-scene

At the user’s disposal are three methods for navigating this expansive virtual environment. He or she may walk around physically within the confines of the SCAPE room to explore the World Local context. The user may also move $W_L$ within $W$ by transforming his or her particular instance of $T_{W_i}^{W}$. The user wears a 5DT DataGlove that influences this transform; he or she can thus point forward or backward to nudge $W_L$ along the his or her line-of-sight in the World Global space.

Alternatively, the user may reposition $W_L$ by placing an identifying marker on the virtual workbench map, as shown in Fig. 12-a. The marker location is calculated relative to the micro-scene by the vision tracker, and the user’s transform $T_{W_i}^{W}$ is then updated to shift the walk-through view accordingly (Fig. 12-b). In this instance, the micro-scene also serves as a WIM navigational metaphor that is conceptually similar to existing interfaces [27,21].

In multi-user settings, the workbench and devices are, as discussed in Section 3.2, “locked” into the same reference frame for all users. Hence while the users share the same physical space corresponding to $W_L$ – the SCAPE room – they can individually explore disjoint virtual locations in the macro-scene city and can then confer via the shared workbench and Widgets.

5.2 Interacting with Widgets

The Aztec Explorer application features the three Widgets discussed in Section 4: CoCylinder, Magnifier, and CoCube. We subjectively evaluate these devices and mention the perceptual issues that arose during testing.

CoCylinder Widget. We used the CoCylinder widget as a means of examining life-size virtual artifacts, namely pieces of pottery, as shown in Fig. 7-b. Due to its simple design and limited range of motion (azimuth rotations only), the Widget required only coarse registration of its real and virtual components to achieve a sense of presence; consequently the CoCylinder was also the Widget least susceptible to user perception problems.

Magnifier Widget. The Magnifier widget allowed an area of the Tenochtitlan map to be observed in high detail. For this application, we found that a field of view of 8° for the Magnifier’s virtual camera provided acceptable results. At this magnification, users quickly located colorful temples and altars in the Magnifier inset (Fig. 1-b). A significantly narrower FOV, with consequently greater magnification, tended to make navigating landmarks difficult, due to the repetitive, disorienting stone patterns in the texture maps.

While accurate registration of the Magnifier’s real and virtual components was not necessary for the device to operate, good registration enhanced the users’ sense of presence of the device as a coherent unit. In our testbed, registration was reasonably good, although some misregistration (<1 cm) was evident when the device was moved to the map periphery, due to the accuracy of the Widget tracking device.

CoCube Widget. For this application, the CoCube widget permitted users to minify large temples for easier inspection (Fig. 11-a) and to read about the structures’ history and architectural significance via the book interface. We specified 14 buildings that could be individually captured via the Widget’s selection mode. We found that a line-of-sight tolerance $\varepsilon$ of 0.005, with a corresponding activation angle of about 6°, yielded a fairly intuitive selection heuristic; significantly larger tolerances were prone to accidental invocation of the selection mechanism.

To toggle to the book interface, we encoded a hand gesture for the 5DT DataGlove. When the user toggled to this mode, he or she was presented with the title of the structure, followed by 5-10 3D pages, depending on the building. Each page housed 1-2 short paragraphs illustrating the history and features of the building, as shown in Figs. 11-b
to 11-d. Due to the limited resolution of our HMPD’s LCDs (640x480), we used larger fonts (>20pt) and limited each page to less than 40 words to improve viewing comfort. Also, the use of DataGlove gestures was cumbersome in this instance, since two hands were required to manipulate the CoCube; the inclusion of interface buttons on the CoCube panels would be much more desirable.

As with the Magnifier, registration of the CoCube’s virtual components with the physical box was fairly accurate, given that we used only simple empirical calibration of the magnetic tracker. When the Widget was held stationary, the mis-registration was typically less than ½ centimeter, and this discrepancy was significantly masked by the CoCube’s occluding borders. While tracker lag typically prohibited users from interacting with augmented objects as quickly as with native physical objects, the effects of lag were most noticeable during CoCube manipulation; brisk movement of the physical box led to significant momentary mis-registration of the real and virtual components, affecting users’ perception of a cohesive augmented object.

6 SUMMARY AND CONCLUSIONS
In this article, we have enhanced the SCAPE collaborative infrastructure by providing tangible interaction via augmented Widgets. More specifically, we have formalized SCAPE’s transformation hierarchy and subsequently used it in the construction of three augmented Widgets, including the CoCylinder, Magnifier, and CoCube. This extended infrastructure is capable of providing multiple participants: (1) functional integration of virtual and physical spaces; (2) a shared workspace with tangible tools and face-to-face interpersonal communication; (3) multiple ways of selecting, examining, coordinating, and sharing synthetic objects via augmented tools.

We are presently conducting a formal user study of the SCAPE system. Namely we are investigating the cooperative effectiveness and ease of use of SCAPE’s bimodal interface and Widgets. We are also addressing more direct ways of coordinating the disjoint views of users in a collaborative setting, when group inspection or conferral is desired. The CoCube currently provides the only direct link between users’ immersive views, and the limited resolution of this device makes it unsuitable for conveying the essence of the macro-scene as a whole.

Concerning SCAPE’s interface modalities, future work should focus on the following: (1) the implementation of more advanced calibration methods for Widget tracking devices; (2) the extension of the CoCube to a more robust information retrieval and interfacing system featuring physical buttons on the faces; (3) the investigation of other application domains. Other interface techniques of ubiquitous computing could be potentially integrated into SCAPE, such that it serves as a comprehensive facility for 3D visualization, interaction, and collaboration.

ACKNOWLEDGEMENTS
We thank the following people for their stimulating discussion and collaboration: Jannick Rolland of the ODA Lab at the University of Central Florida, Narendra Ahuja of the Vision and Robotics Lab at the University of Illinois at Urbana-Champaign, Frank Biocca of the M.I.N.D Lab at Michigan State University, and Susan Warshauer of the Center for Literary Computing at West Virginia University. We also specially acknowledge 3M Inc. for generously supplying the retro-reflective film. This paper is based on work supported by National Science Foundation Grants ITR 0083037 and ITR 0301874.

REFERENCES


