Putting the Surface in Context

Final Report 2009–2010

Microsoft Research External Research and Microsoft Surface Product Group *Going Beneath the Surface* Program

Steven Feiner (PI), Nicolas Dedual, Jacquilene Jacobs,
Ohan Oda, Charlie Robbins, Mengu Sukan, Sean White

Columbia University
Dept. of Computer Science
500 W 120th St., 450 CS Building
New York, NY 10027
feiner@cs.columbia.edu
212-939-7083

Figure 1. (Left) Surface botanical field guide in use by visitors to the exhibit *Since Darwin: The Evolution of Evolution* at the Smithsonian. (Right) Virtual models of the World Financial Center overlaid on the Surface, as viewed through tracked video–see-through eyewear.

1. Introduction

Columbia's Computer Graphics and User Interfaces Lab performed research that addressed two main ways in which the Microsoft Surface could be used in context with other computing devices and systems.

First, we explored how the Surface could be used with existing vision-based recognition algorithms, ranging beyond those supported by existing Surface software. In this research, we developed Surface-based members of the LeafView family of electronic field guides for botanists [Agarwal et al., 2006]. LeafView, which was developed with colleagues at Columbia, U Maryland, and the Smithsonian, combines image-based recognition of plant leaves with navigation of a database of botanical specimens. We investigated some of the difficult problems involved in adapting the previous prototypes, which used Tablet PCs and UMPCs with conventional cameras (both built-in and wireless), to instead use infrared imaging and interaction on the Surface tabletop [White & Feiner, 2010]. Complementing this research, we took advantage of a unique opportunity to develop for exhibit at the Smithsonian, a stand-alone Surface-based application for botanical education, shown in Figure 1 (left). Starting in June 2010, the resulting system appeared as part of the exhibit *Since Darwin: The*

Evolution of Evolution (http://www.mnh.si.edu/exhibits/darwin/) at the Smithsonian National Museum of Natural History, and was the first Surface application to appear at the Smithsonian.

Second, we explored how the Surface could be used along with other displays to create a heterogeneous information space in which users can benefit from the complementary interaction and display capabilities of each device. We concentrated on the use of head-tracked, see-through eyewear to present first-person 3D augmented reality imagery, complementing the 2D imagery displayed on the Surface, as shown in Figure 1 (right). We experimented with a variety of six-degree-of-freedom tracking techniques, ranging from using the Surface to display projected tracking markers, to framing the Surface with a printed array of markers, to employing an existing infrared-camera–based tracking infrastructure. To demonstrate the advantages of the concept, we addressed several domains, ranging from a CAD application in which users could create and edit simple 3D models, to an urban visualization application in which users could explore parts of Manhattan, using a combination of Bing Maps and 3D Manhattan building models made available to us by Microsoft Research.

In the following sections we report on each of these lines of research.

2. Botanical Field Guides for Surface

This research explored how the Surface could be used with vision-based recognition algorithms [Belhumeur et al., 2008] originally developed by our colleagues for ultramobile PCs (UPMCs) and Tablet PCs. Our domain was electronic field guides for botanists, which we had participated in developing in collaboration with researchers at Columbia, University of Maryland, and the Smithsonian [Agarwal et al., 2006]. In the workflow used for the LeafView family of experimental field guides, the user prepared a leaf by flattening it against a uniformly colored piece of paper. The user then photographed the leaf with the built-in camera of the UMPC or a stand-alone camera wirelessly interfaced to the Tablet PC. Releasing the shutter triggered the system to initiate segmentation of the leaf image and subsequent matching. When this was complete, the user was able to interact with the ordered set of potential matches returned by the system. The challenge we set for ourselves was to replace this workflow with one that would be more natural for the Surface.

Surface Processing Workflow

We began our exploration by building a system that used the LeafView codebase for recognition and local data store for images. Our initial approach to capturing the leaf image was to replace the *explicit* manual "shutter release" of the camera-based LeafView systems with an *implicit* processing trigger: We wanted the act of placing a leaf on the Surface to trigger image capture using the Surface's infrared cameras, segmentation, and searching, which would be visually indicated by presentation of animated feedback around the leaf. Frames from a video of our initial prototype are shown in Figure 2.

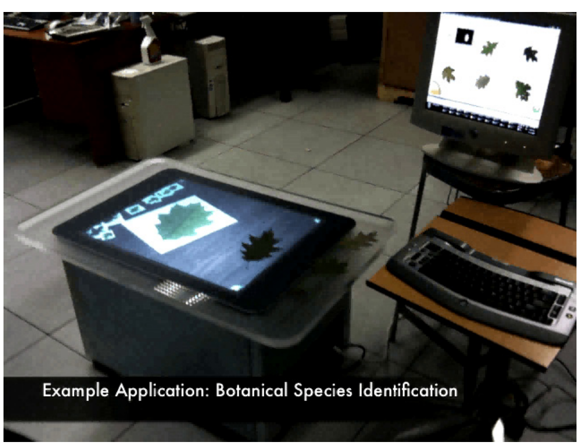
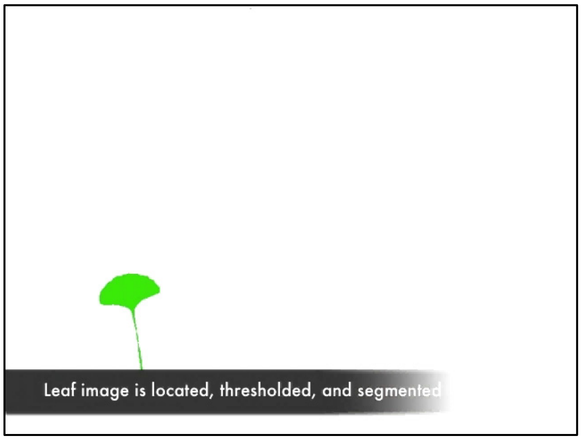
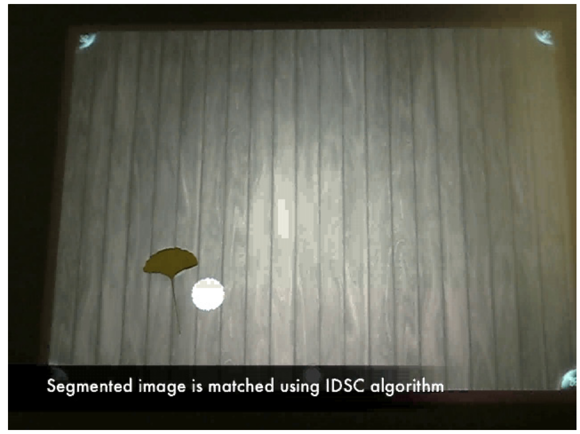
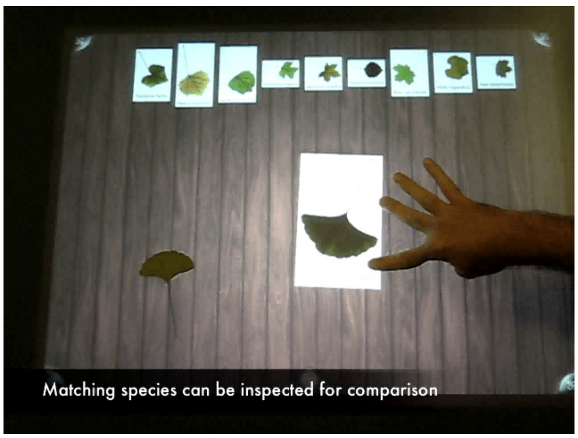


Figure 2. Initial botanical field guide for Surface. (a) User interface. (b) User places a leaf on the Surface, implicitly triggering processing by existing vision-based algorithms.

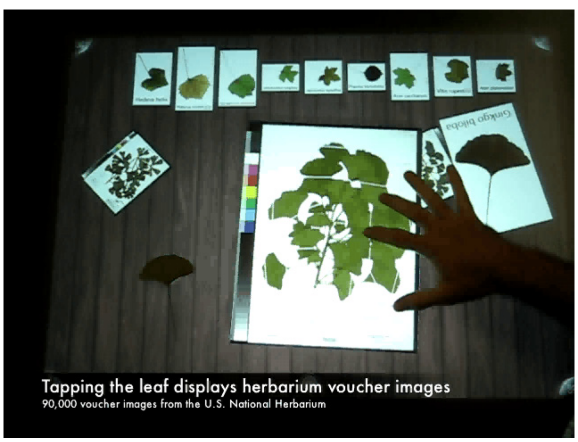




(c) Leaf image is captured from Surface infrared cameras and segmented. (d) Segmented image Is matched to specimens in database.



(e) Images of potential matching species are displayed. (f) User inspects species images.



(g) Voucher images of preserved specimens in the US National Herbarium. (h) High-resolution images of species.

Figure 2 (cont). Initial botanical field guide for Surface.

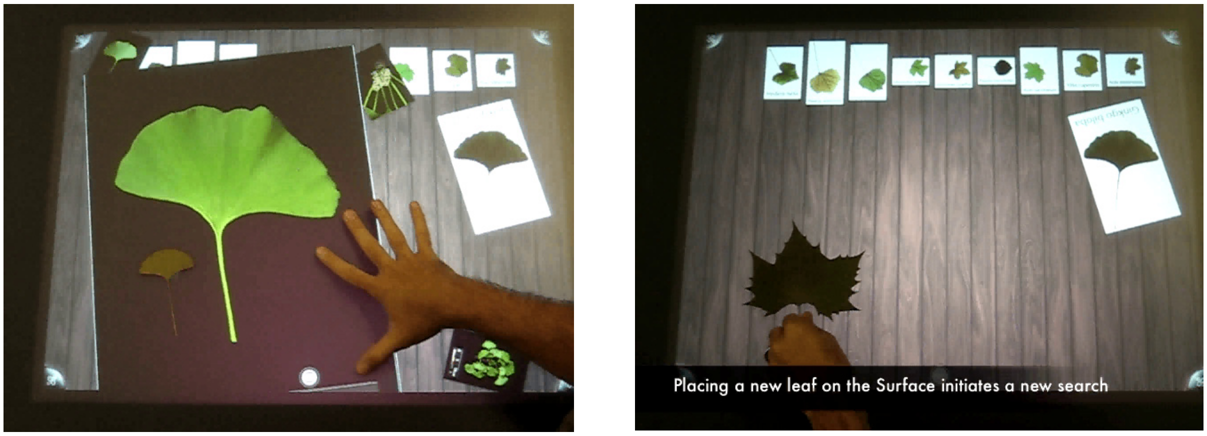


Figure 2 (cont). Initial botanical field guide for Surface. (i) User inspects high-resolution images. (j) User places a new leaf on the table to initiate a new search.

In developing our initial prototype, we started by using the recognition of a blob contact as a trigger to capture the Surface's infrared image for processing. This raises the following issues, noting that they appear to apply to objects in general, rather than to leaves in particular:

- *When* to try to identify the object. As an object approaches the Surface, its projection changes shape from the time it is first seen by the Surface's infrared cameras.
- Getting the *right* object. There are often multiple contacts per object and multiple objects per contact. Because the user can hold an object when placing it, rather than dropping it, the hand and object can be mistaken for a single object or appear as multiple objects.
- *Changing* IDs. An object placed on the Surface can change ID when touched or moved, whether accidentally or intentionally.
- *Nonplanar* objects. Objects that are not planar, such as a curling leaf, can result in multiple points of contact and changing IDs. They can also result in incorrect segmentation when parts of the object that are not in contact are not viewed by the cameras.
- *Unrecognized* objects. Some small lightweight objects, including leaves, do not trigger contact.
- *Small* objects. Sufficiently small objects are recognized as fingers, rather than as blobs.

Since we wanted to condition image capture based on existing support for detecting a blob contact, we attempted to address these initial problems through a set of heuristic approaches:

- Waiting for *stabilization*. When a blob is detected, we monitor size, position, and contact ID, waiting until size and position stabilize within an experimentally determined range for a constant ID. This helps avoid false positives at the expense of introducing a wait. Unfortunately, this creates the possibility of false negatives, in which a real object on the surface that appears to be visually stable to the user, is seen as continually changing in size by the Surface and never satisfies our stabilization criteria. (Here, we note that these false negatives could be avoided by including a user interface element to explicitly initiate processing.)
- Supporting *movement*. We wished to allow a user to move an object that does not maintain steady contact with the surface, without it spuriously retriggering processing because its ID changes. One approach would be when the ID of a recognized object's blob disappears and a new blob ID appears, to attempt to determine whether the old and new blobs are sufficiently similar in location and shape to warrant classifying them as the same object. Rather than fol-

lowing this route, we instead opted to include a user interface element to explicitly cancel spurious retriggering.

- Supporting *nonplanar* objects. As noted above, nonplanar objects, and especially lightweight ones, can cause multiple problems, including changing IDs, multiple IDs, no ID, and segmentation that misses parts of the object that are not adequately imaged. (In contrast, nonplanarity in the original LeafView prototypes, which used conventional cameras, would typically result in segmented images that were too large because shadows cast by curled parts of a leaf would be included in the segmented shape.) Inspired by the plant presses used by botanists to flatten specimens, we addressed this problem through the use of a flat piece of non–infrared-reflecting plastic that could be laid over the leaf to flatten it.

Multi-User Interaction

While these techniques improved the processing workflow, they introduced a wait and were still not sufficiently reliable. Therefore, we decided to use an explicit trigger model as we moved on to create the collaborative multi-user version of the Surface application shown in Figure 3. For the multi-user system, we decided that a user would first specify a bounding box for segmentation by using a user interface element to explicitly indicate a rectangular area within which the leaf would then need to be placed. When the user is satisfied with their placement of the leaf, they trigger processing with a user interface element (Figure 3a), initiating animated feedback. The segmented leaf is then displayed, allowing the user to terminate and restart processing if they are not satisfied with the result. When searching has completed, the leaf is surrounded by information about potential matches (Figure 3b). Because of the use of explicitly specified placement areas and triggers, we can reliably support multiple users simultaneously interacting with multiple leaves.

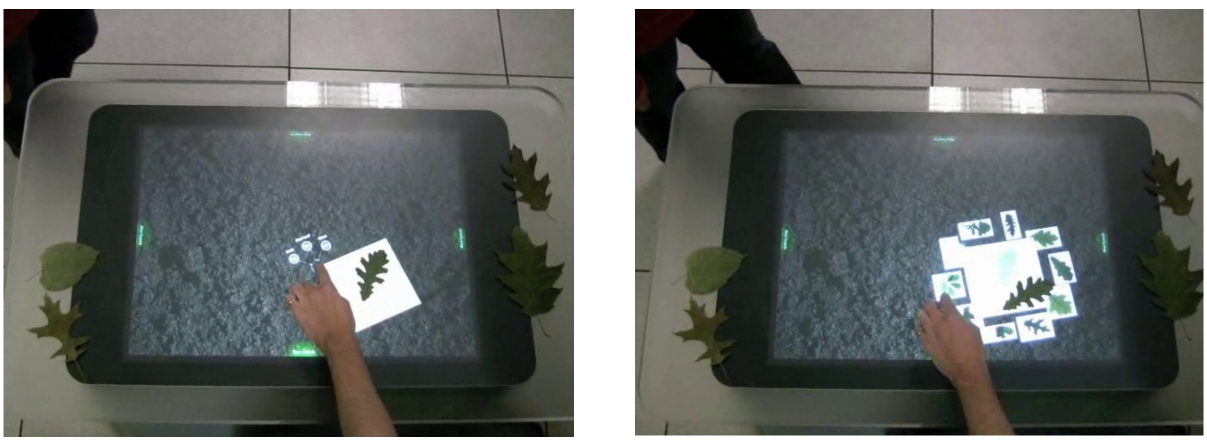


Figure 3. Collaborative multi-user version of the botanical field guide for Surface. (a) User specifies a bounding box for the leaf, places the leaf inside it, and triggers processing by explicitly selecting a user interface element. (b) System captures image of leaf, segments it, displays segmented image in green for feedback, performs matching, and surrounds leaf with images of potential matches.

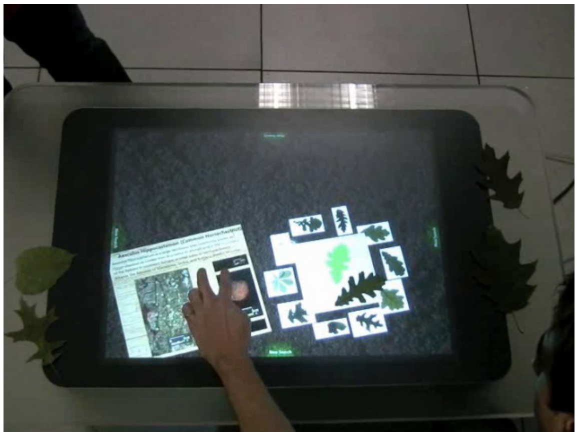
 

Figure 3 (cont.). Collaborative multi-user version of the botanical field guide for Surface. (c) User selects one of the potential matches, whose image is expanded into an information sheet, at the left, containing text and a scrolling panel of images. (d) Additional users can independently create bounding boxes, initiate processing of additional leaves, and explore their results.

Software Architecture

While our initial design based on the LeafView codebase provided a useful demonstration of concepts, it had several drawbacks. LeafView was written for Windows XP, using compiled Matlab, making it unstable on later versions of Windows, and difficult to install on new systems. It required significant CPU resources that can affect interaction on the Surface, and the local data store was over 4 GB, making it difficult to share the application. In addition, the combined application and local store did not provide a generalized service solution to identification and interaction.

To address these issues, when developing our multi-user system we integrated a JSON-based API to a separate identification and data server that is also used for the mobile identification devices. This made identification faster and cleaner, and potentially more generalizable within the API. At the same time, we also developed a service-oriented architecture, shown in Figure 4, implemented with Microsoft Unity and Prism. Unity performs dependency injection, while Prism is responsible for Event Aggregation (i.e., local message queue) and UI composition using the RegionManager. Presentation logic is encapsulated into attached behaviors whenever possible. Attached behaviors take advantage of attached properties in WPF to add UI functionality to an element without modifying its code directly. They also can be used in Microsoft Expression Blend (the WPF/Silverlight design tool).

We released the following open source software resulting from this effort under an MIT license:

- Remote logging service: http://github.com/indexzero/exceptiony
- Toolkit of WPF behaviors/services for logging UX events: http://github.com/indexzero/ux-logging-toolkit
- Behaviors for capturing and working with raw input on Surface: http://github.com/indexzero/surface-raw-input

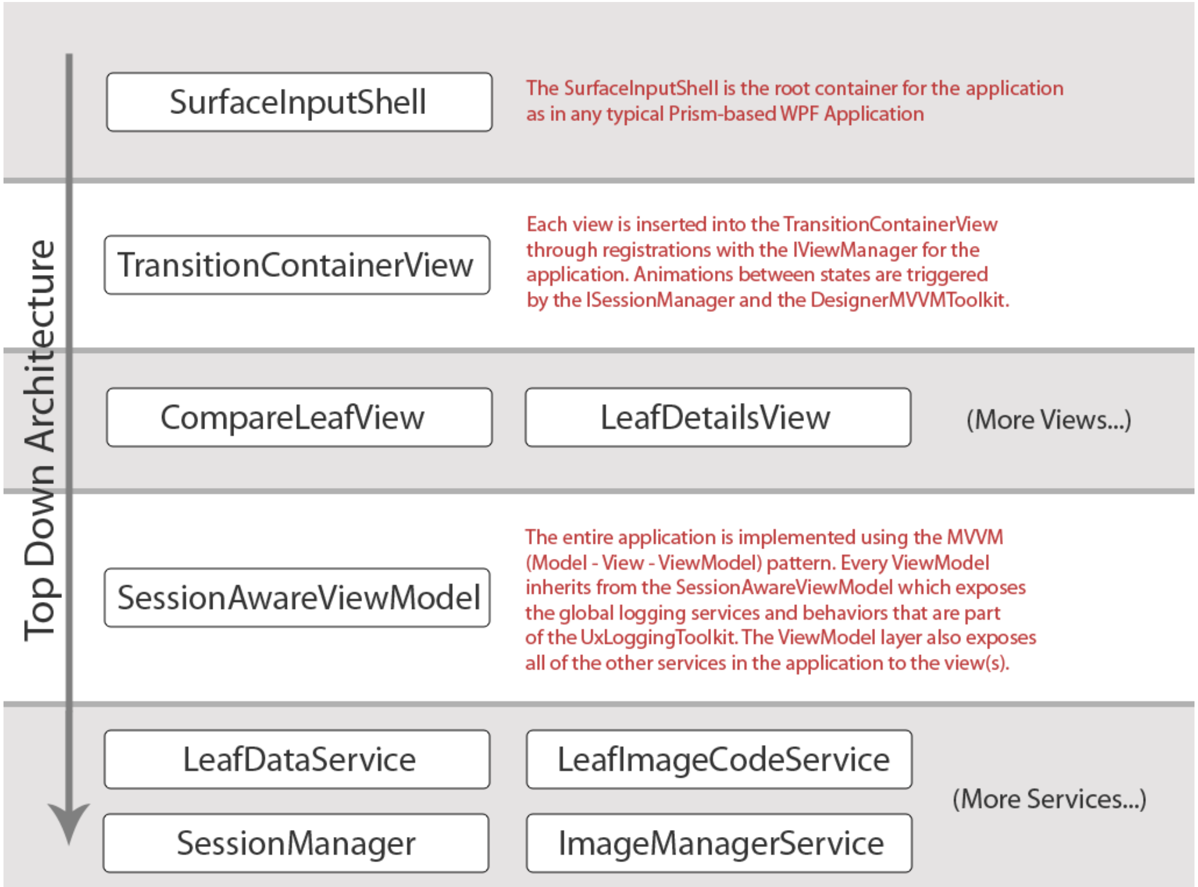


Figure 4. Software architecture of collaborative Surface electronic field guide.

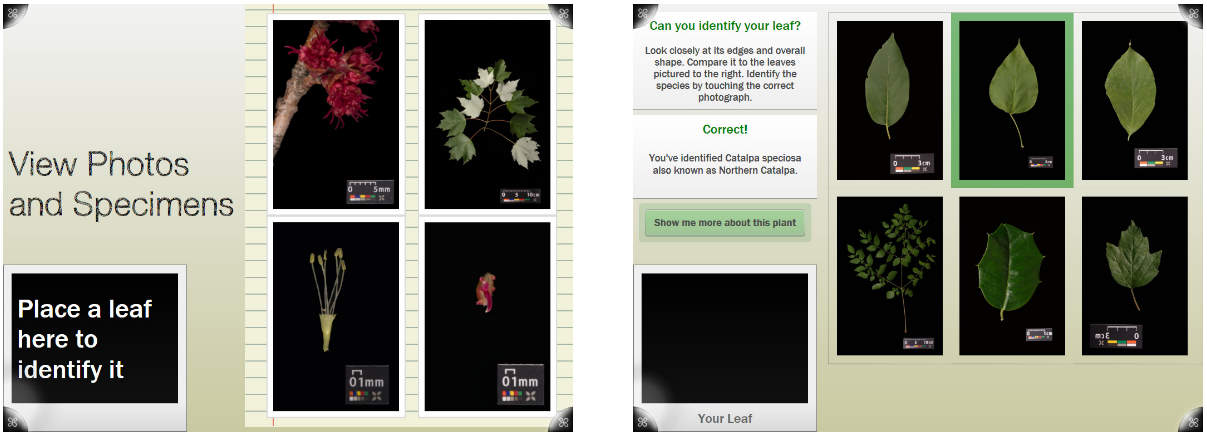
Smithsonian Exhibition

In late 2009, we were given the opportunity to develop a Surface-based educational application that could be deployed in an exhibit at the Smithsonian Institution. We had shown our early explorations of LeafView on the Surface to our colleagues in the Department of Botany at the Smithsonian Institution, and they agreed to work with us on designing a version of the system to appear in *Since Darwin: The Evolution of Evolution* (http://www.mnh.si.edu/exhibits/darwin/), an exhibit at the Smithsonian National Museum of Natural History.

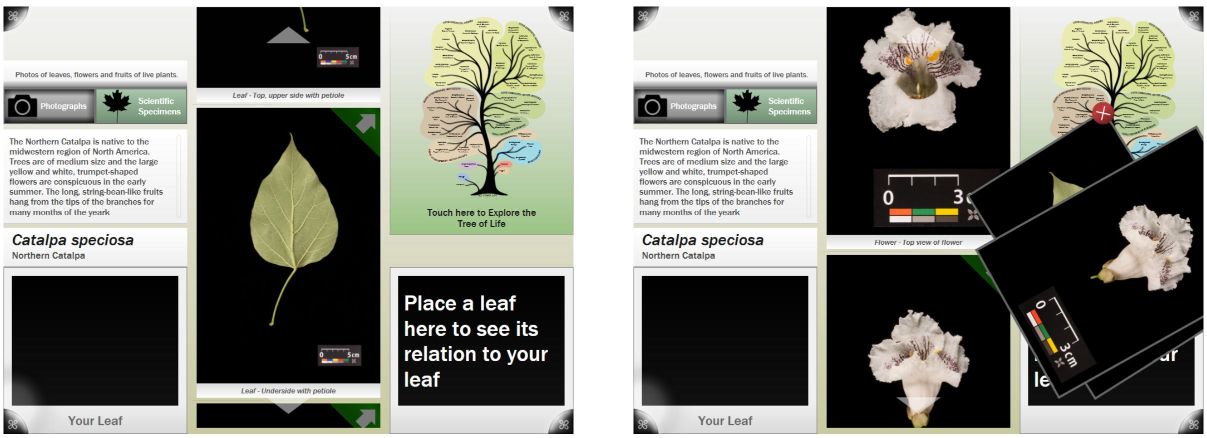
Over a period of six months, a ten-person team spanning Columbia's Computer Graphics and User Interfaces Lab and the National Museum of Natural History (Research, IT, Exhibits, and Education) worked to iteratively design and develop the application, which was deployed in June 2010. We decided to make it a single-user experience with opportunities for intimate groups to cooperate. To address durability of the leaves, we used laser-cut plastic leaf templates instead of real leaves. To address speed and robustness of the segmentation and matching, we replaced the custom vision algorithms used in the LeafView prototypes with Surface tag-based matching. We added a species guessing game to encourage interaction and integrated an animated representation of the "tree of life" (representing botanical taxonomy) to fit the exhibit theme. Figure 5 shows the system as it appeared in the exhibit, while Figure 6 shows frames captured during an interactive session.

In addition to the Smithsonian exhibit, we presented this work at our demonstration of electronic field guide user interfaces at *CHI 2010* in Atlanta (although without a Surface), demonstrated it live in the Microsoft booth at the *ISTE 2010 National Education Computing Conference* in Denver, and installed it for attendees to try at the *2010 MSR Faculty Summit*.

Figure 5. Surface botanical field guide in use by attendees at *Since Darwin: The Evolution of Evolution,* an exhibit at the Smithsonian National Museum of Natural History. Surface is mounted on custom-built base. Each plastic leaf template has a Surface tag on the bottom and is tethered to the panel at the front.



(a) Initial attract screen. (b) User has placed a leaf template at the lower left and selected the correct matching leaf (highlighted in green).



(c) User requests additional information about the plant (Catalpa speciosa) and is presented with text and high-resolution images. (d) User scrolls through images and selects some to explore in detail.

(e) User replaces the leaf template at the lower left with a different one to learn about another plant (Liquidambar styraciflua) and chooses to examine digitized scientific specimens from the US National Herbarium. (f) Placing a second leaf template (Gingko biloba) at the lower right allows the user to view an animation of the paths through the tree of life to the two species.

Figure 6. Screen shots of Surface botanical field guide developed for *Since Darwin: The Evolution of Evolution,* an exhibit at the Smithsonian National Museum of Natural History. Note: Images are captured directly from the Surface framebuffer for clarity and do not show the user and leaf templates.

3. Heterogeneous Information Spaces

Computing displays and devices present an increasing diversity of form and deployment. Some are hand-held, such as smartphones, personal media players, UMPCs, and Tablet PCs. Others reside on the desk, and include PCs with single and multiple monitor configurations. Some cover tabletops, walls, or floors. Others are head-worn. How can these heterogeneous displays and devices complement each other to create an effective environment in which to work, play, and learn?

In this part of our research, we explored the design and development of hybrid user interfaces [Feiner and Shamash, 1991] that coupled the 2D multiuser tabletop user interface afforded by Microsoft Surface with the 3D user interface of head-tracked video–see-through eyewear. Essentially, we extended the 2D planar interaction and display area of the Surface with the 3D volume above and around it to support a first-person augmented reality view. Our work was conducted using our Goblin XNA infrastructure for 3D user interfaces and games, built on top of XNA Game Studio 3.0 and distributed at http://goblinxna.codeplex.com.

We note that there are interesting complementary differences between the passive paper surfaces illuminated by Second Light [Izadi et al., 2008] and the active devices we used. For example, while Second Light has the advantage of using (literally) paper-thin, passive display surfaces, they are illuminated at the resolution of the Surface projector, can be easily blocked from illumination when many users interact by holding them above the table, and require the use of a projector-based Surface. In contrast, active devices can have much higher resolution (pixel density), make no assumptions about the Surface technology, can store, provide, and acquire information associated with their owner, and can support private viewing in the case of a head-worn display; however, they currently weigh significantly more and have a much smaller display area.

Initial Exploration: CAD and Games

For our first exploration, we created a simple CAD system in which the 2D Surface shows semi-transparent 2D footprints of simple geometric models and other parts of the user interface, while the 3D head-tracked eyewear presents the full 3D models, positioned over their footprints on the Surface. This makes it possible for a user wearing the eyewear to view the 3D geometric models in augmented reality, as if they were physically located on the Surface. Multi-touch gestures are used to create and edit the models. The 3D eyewear (Vuzix iWear VR920 with CamAR) is tracked with the VTT ALVAR optical tracking package, using markers projected on the Surface. The markers are viewed by a camera built into the eyewear, which provides the view of the real world. Figure 7 shows several frames from a video (http://www.youtube.com/watch?v=ym1xaRrfgTk) of this system. (Note that the 3D models extend up and beyond the edge of the Surface screen, which would not happen if the head-tracked graphics were displayed on the Surface itself.)

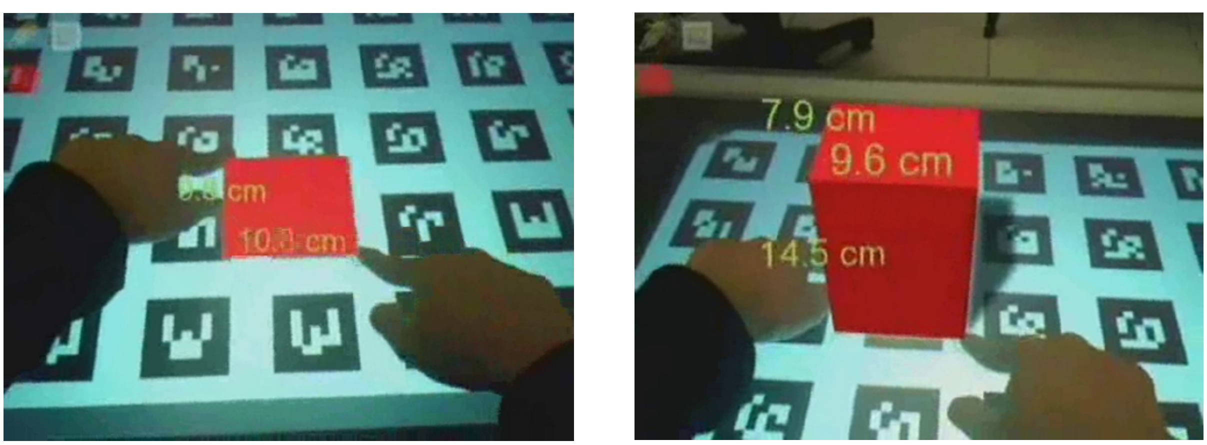


Figure 7. Simple CAD prototype using tracked video–see-through eyewear with Surface. (a) Specifying the base of a rectangular prism with two fingers. (b) Extruding the object into 3D using a thumb gesture.

Figure 7 (cont.). Simple CAD prototype using tracked video–see-through eyewear with Surface. (c) Changing the color of an object. (b) Translating an object on the Surface.

An important aspect of this hybrid user interface is that users who view the Surface without eyewear still see the object footprints and construction interface, which are spatially registered with the models. Those users can also interact with the Surface and can point within the same space viewed by users with eyewear. For example, Figure 8 (a) shows the display presented on the Surface, including object footprints and the 2D user interface being used to specify the base of an irregular polygon, such as the one at the left of Figure 7 (c–d). Since footprints are occluded by their 3D objects as seen by users with eyewear, we experimented with using the footprints to present information to other users without cluttering the view of users with eyewear. For example, Figure 8 (b) shows how each footprint can be used to display a 3D rendering of the footprint's object from a canonical view.

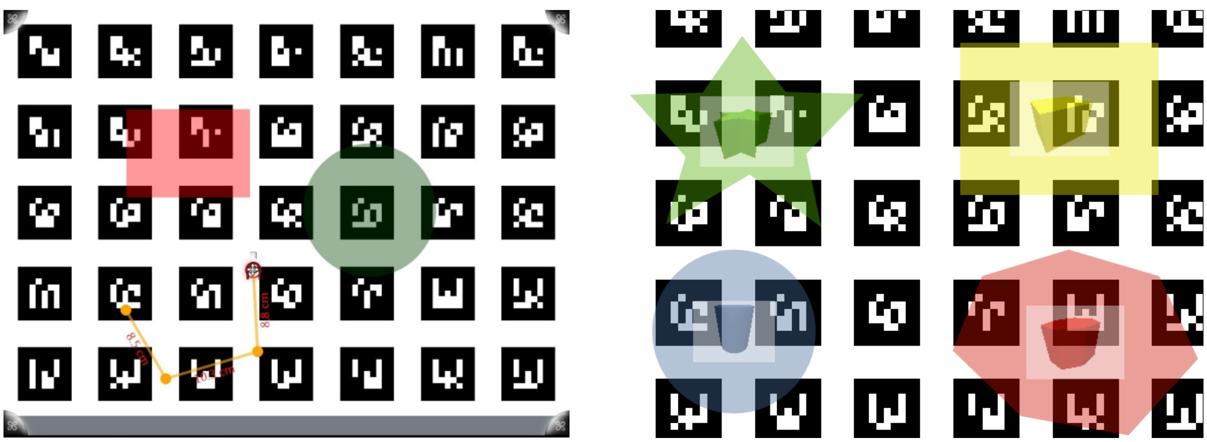


Figure 8. Surface display of the simple CAD prototype. (a) Footprints of objects and user interface for specifying an irregular polygon. (b) Each 2D footprint can display a rendering of its associated 3D object.

Figure 9 shows a different treatment of the Surface graphics, used in a study for a multi-player role-playing game: Abstract 2D symbols displayed on the Surface correspond to simple 3D models seen through the eyewear.

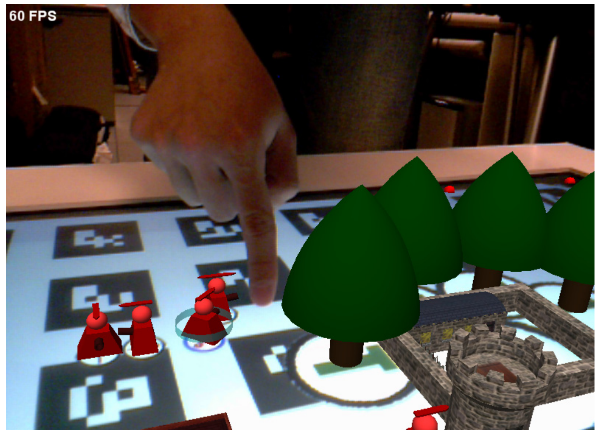


Figure 9. Study for a multi-player role-playing game. 3D graphics displayed on head-tracked video–see-through eyewear are complemented by 2D abstract graphics on the Surface.

Urban Visualization

One domain that we found especially attractive is that of *urban visualization*, creating a vivid, manipulable representation of an urban environment that can be simultaneously explored and modified by multiple users. We developed a prototype that combines map data acquired from the Bing Maps API with 3D building models spanning the lower part of Manhattan up to 34th Street, provided by Bill Chen and Johannes Kopf (Microsoft Research).

We began by creating a WPF application, running on the Surface, which uses gesture recognition controls to translate, scale, and rotate the map data. Our application transforms the Latitude, Longitude, Altitude (LLA) coordinates associated with the corners of the Surface into Earth-Centered Earth-Fixed (ECEF) coordinates. The ECEF coordinates are then used to look up buildings within the bounded area. The ECEF coordinates of the buildings are further transformed into a local East, North, Up (ENU) coordinate system, projecting the ECEF coordinates onto a tangential plane. We decided that this simplified approach would be suitable for our explorations, since the topography of lower Manhattan is planar to a first approximation.

In our initial prototype, we tracked the 3D position and orientation of the eyewear using the ALVAR optical marker tracking toolkit, integrated with Goblin XNA. Instead of displaying markers directly on the Surface, as we had done in the CAD and game prototypes, we surrounded the screen with a precisely aligned foamcore-backed printed frame of markers. Figure 10 shows this approach being used to view part of Manhattan's World Financial Center. As the user translates, scales, and rotates the maps displayed on the Surface, the building models are transformed correspondingly.

While the tracking frame provided an inexpensive and accurate way to determine the 3D position and orientation of devices relative to the Surface, it required that the eyewear always look at the Surface to be tracked. This is not a problem when the objects do not rise significantly above the Surface, as in our earlier applications. However, our urban test domain is filled with many tall buildings. If we choose a view in which there are relatively few city blocks represented on the Surface, then requiring that the eyewear must always see the bases of the buildings is severely constraining.

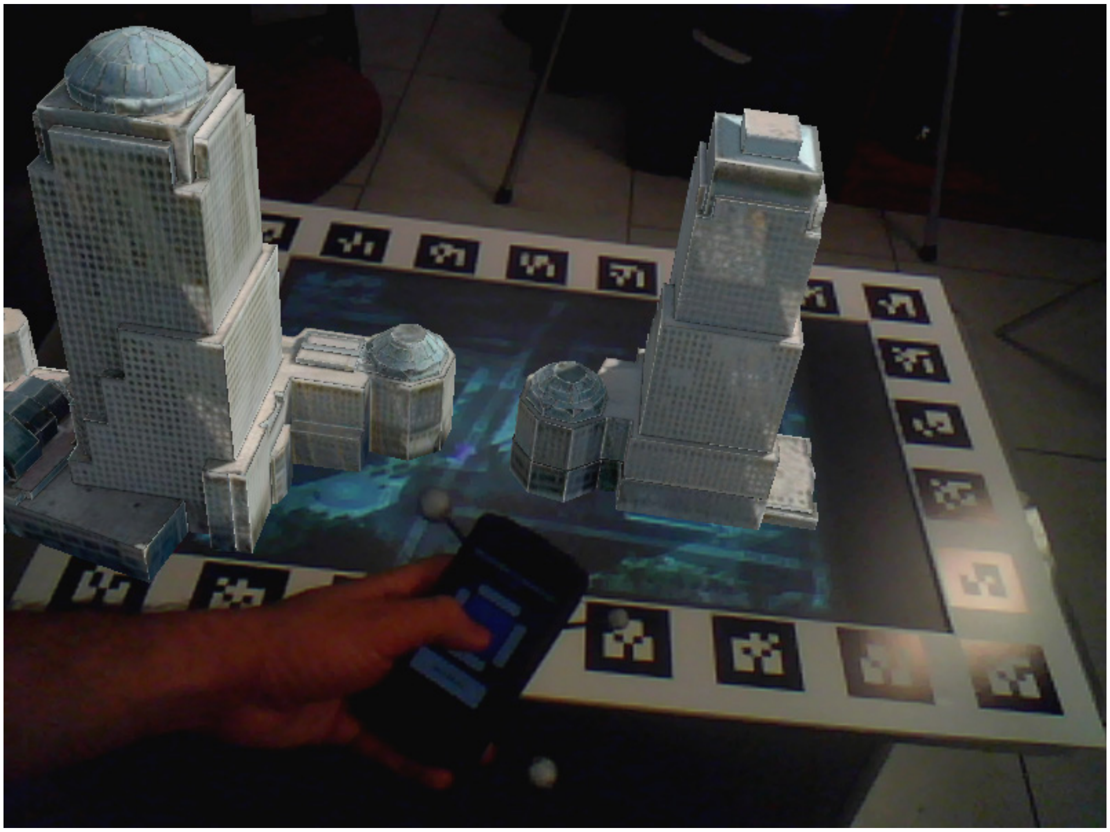


Figure 10. Urban visualization prototype. 3D virtual building models of part of the World Financial Center are presented on head-tracked eyewear and overlaid on the buildings' footprints displayed on Surface using the Bing Maps API. Hand-held smartphone is also tracked relative to the Surface. (Building models courtesy of Bill Chen and Johannes Kopf, Microsoft Research.)

To address this, we replaced the optical marker frame with a NaturalPoint OptiTrack infrared tracking system. This approach uses a cluster of ceiling-mounted infrared-sensitive cameras aimed into the volume above and around the Surface. The 3D position and orientation of the Surface are known relative to the OptiTrack cluster. A rigid marker configuration of small retroreflective spheres is fixed to the eyewear and illuminated by a ring of infrared LEDs surrounding the lens of each camera. Each camera thus sees a 2D pattern formed by the reflections of the spheres in its frustum. The 2D patterns seen by multiple cameras can be used to solve for the 3D position and orientation of the configuration of spheres, and hence of the eyewear to which the configuration is attached. Since the Surface emits infrared light, we needed to locate the OptiTrack cameras carefully and use facilities in the OptiTrack interface to ignore infrared light from the Surface. Since the OptiTrack cameras emit infrared light, we also had to position the cameras to keep them from affecting the Surface.

Figure 11 shows frames from a video (http://www.youtube.com/watch?v=zaXv570g2YI) that uses this tracking system to view a portion of Manhattan's Flatiron District. (Note: This video was recorded live in July 2010 in a temporary installation outside our lab, using a horizontal LCD panel running the Microsoft Surface Simulator as a stand-in for our lab's Microsoft Surface, and was presented at the *2010 Microsoft Research Faculty Summit.*)

We also used our OptiTrack tracking infrastructure to experiment with how a smartphone handset, shown in Figure 10, can be tracked using an affixed configuration of marker spheres, and used to interact with the urban database to select and query information about models in the volume around the Surface.

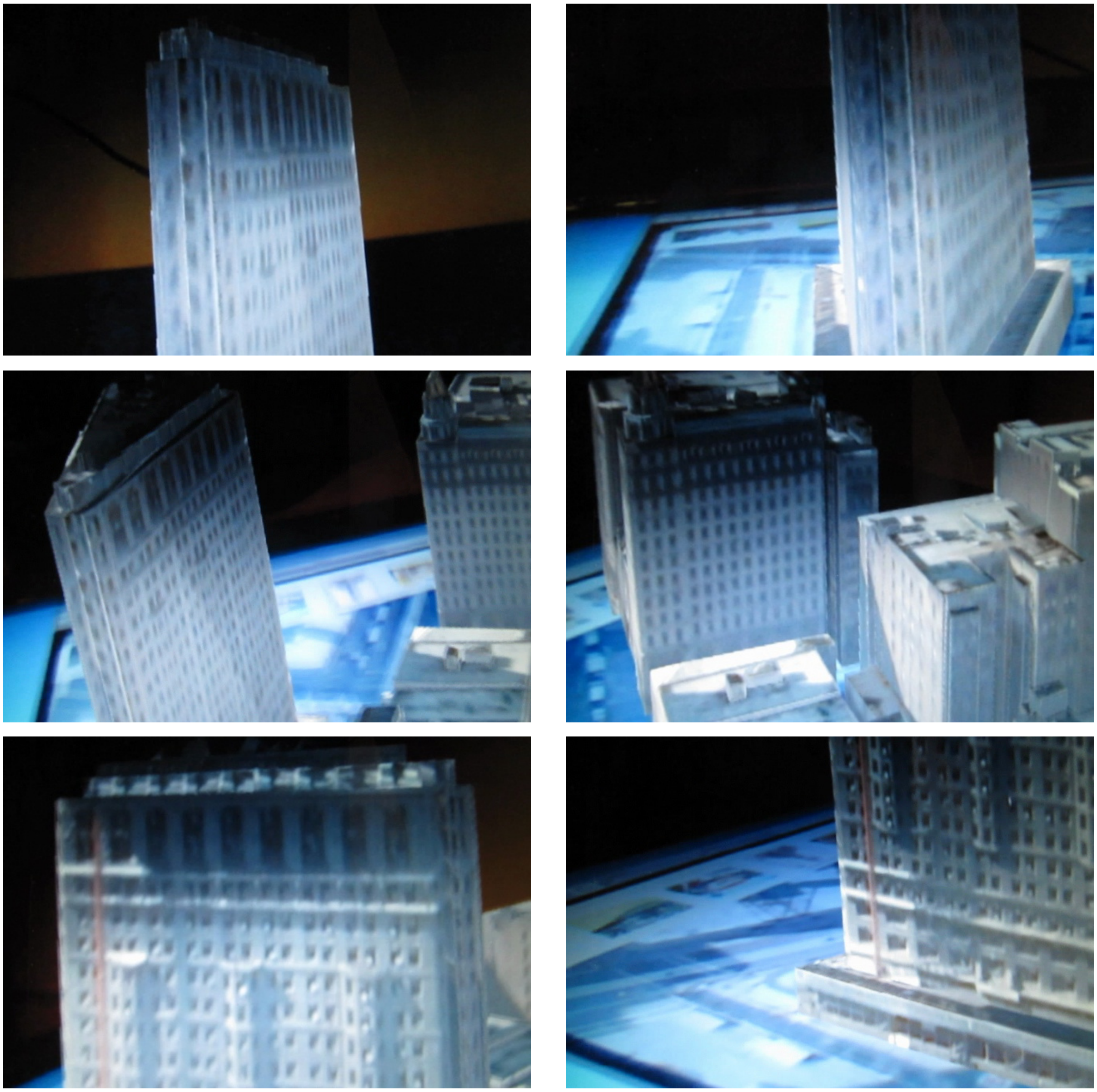


Figure 11. Urban visualization prototype. Frames from a video showing 3D virtual models of buildings in Manhattan's Flatiron District presented on head-tracked eyewear. The models are overlaid on the buildings' footprints displayed on Surface using the Bing Maps API. Note that the image at the upper left could not have been tracked using the marker frame of Figure 10, since the Surface is not visible. (Building models courtesy of Bill Chen and Johannes Kopf, Microsoft Research.)

We are currently continuing to build on the work done for this project, to create an infrastructure that supports interaction across a wide range of tracked stationary and mobile devices. We are especially interested in integrating urban infrastructure models with relevant databases (e.g., Building Information Models and energy). We would also like to incorporate whole body and environment sensing technologies such as Kinect.

Dissemination

Our Surface botanical field guide was installed in the exhibit Since Darwin: The Evolution of Evolution (Smithsonian National Museum of Natural History, Washington DC, June–July 2010). In addition, live demonstrations of this work were presented in the Microsoft booth at the *ISTE 2010 National Education Computing Conference* (Denver, CO, June 27–30, 2010) and at the *2010 MSR Faculty*

Summit (Redmond, WA, July 12–13, 2010). The application was also presented (but not demonstrated, since a Surface was not available) at our Media Showcase exhibit and publication at *CHI 2010* (Atlanta, GA, April 10–15, 2010).

Open source (MIT license) software resulting from this effort includes:

- Remote logging service: http://github.com/indexzero/exceptiony
- Toolkit of WPF behaviors/services for logging UX events: http://github.com/indexzero/ux-logging-toolkit
- Behaviors for capturing and working with raw input on Surface: http://github.com/indexzero/surface-raw-input

During the course of the grant, we presented our work at a number of invited talks, including the Distinguished Lecture Series, University of Aizu, Aizu-Wakamatsu, Japan, November 16, 2009; a keynote address at *WARM 2010* (*Fifth Winter Augmented Reality Meeting*), Technische Universität Graz, Graz, Austria, February 24–25, 2010; *eComm America 2010* (*Emerging Communications Conference and Augmented Reality Conference*), San Francisco, CA, April 19–21, 2010; *are 2010* (*Augmented Reality Event*), Santa Clara, CA, June 2–3, 2010; a keynote address at *Trends and Prospects of 3D*, AIIT (Advanced Institute of Information Technology), Ewha University, Seoul, Republic of Korea, July 3, 2010; and Nokia Research Center Hollywood, Santa Monica, CA, August 6, 2010.

Acknowledgments

This work was funded by Microsoft Research External Research and the Microsoft Surface product group, through the *Going Beneath the Surface* grant program. We wish to thank John Nordlinger, Lee Dirks, Daniel Wigdor, and Donald Brinkman for overseeing the program. Bill Chen and Johannes Kopf of Microsoft Research graciously provided the 3D building models used in our urban visualization domain.

Work on the Surface electronic field guide was performed by Sean White, Charlie Robbins, and Jacquilene Jacobs. Charlie Robbins and Sean White developed the application that was fielded at the Smithsonian National Museum of Natural History. The National Science Foundation funded the work on the original hand-held versions of the electronic field guide, including digitization of the botanical species imagery reused in our Surface applications, through NSF Grant IIS-03-25867. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the official views, opinions, or policy of the National Science Foundation (NSF). The Smithsonian Institution developed the design of the “tree of life” representation used in the Surface field guide.

Exploration of how the Surface could be combined with head-tracked eyewear for simple CAD and game domains was done by Ohan Oda. Mengu Sukan developed support for using the 3D building models in our work. Nicolas Dedual developed our urban visualization infrastructure.

References

1. Agarwal, G., Belhumeur, P., Feiner, S., Jacobs, D., Kress, W.J., Ramamoorthi, R., Bourg, N., Dixit, N., Ling, H., Mahajan, D., Russell, R., Shirdhonkar, S., Sunkavalli, K., and White, S. First steps toward an electronic field guide for plants. *Taxon*, 55(3), August 2006, 597–610.
2. Belhumeur, P., Chen, D., Feiner, S., Jacobs, D., Kress, W., Ling, H., Lopez, I., Ramamoorthi, R., Sheorey, S., White, S., and Zhang, L. Searching the world's herbaria: A system for visual identification of plant species. *Proc. ECCV 2008* (*European Conf. on Computer Vision*), Part IV, LNCS 5305, Marseille, France, October 12–18, 2008, 116–129.
3. Feiner, S. and Shamash, A. Hybrid user interfaces: Breeding virtually bigger interfaces for physically smaller computers. *Proc UIST 1991* (*ACM Symp. on User interface Software and Technology*), Hilton Head, SC, November 11–13, 1991, 9–17.

4. Izadi, S., Hodges, S., Taylor, S., Rosenfeld, D., Villar, N., Butler, A., and Westhues, J. Going beyond the display: A surface technology with an electronically switchable diffuser. *Proc. UIST 2008* (*ACM Symp. on User interface Software and Technology*),Monterey, CA, October 19–22, 2008, 269–278.
5. White, S. and Feiner, S. Exploring interfaces to botanical species classification. *CHI 2010 Extended Abstracts*, Atlanta, GA, April 10–15, 2010, 3051–3055.